NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

DECISION SUPPORT DATABASE SYSTEM
FOR HELLENIC NAVAL PERSONNEL MANAGEMENT

by

Antonios Makris

September 1988

Thesis Co-Advisors: Vincent Y. Lum
John B. Isett

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION Unclassified	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School — 6b OFFICE SYMBOL (If applicable) Code 52	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION — 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO — PROJECT NO — TASK NO — WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

DECISION SUPPORT DATABASE SYSTEM FOR HELLENIC NAVAL PERSONNEL MANAGEMENT

12 PERSONAL AUTHOR(S)

Makris, Antonios K.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM ______ TO ______	1988 September	205

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Goverment.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Database System; Computerized Personnel Management System Decision Support System, Theses, (SDW)

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The Naval Officers Personnel Management System is a very complex system especially inside the Fleet Command. Managing the system manually is neither effective nor efficient in supporting the decision makers.

This thesis proposes a method to use a computer based information processing system to help decision makers in scheduling the assignment of officers to warships during the annual assignment process, as well as in other functions concerning personnel management. The thesis presents a decision support database system for the Naval Officers Management Staff.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL Prof. V. Y. Lum, Major J. B. Isett	22b TELEPHONE (Include Area Code) (408) 646-3091, 646-2464 — 22c OFFICE SYMBOL Code 52Lu, 54Is

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

U.S. Government Printing Office: 1986—606

UNCLASSIFIED

Approved for public release, distribution unlimited

Decision Support Database System for Hellenic Naval Personnel Management

by

Antonios Makris
Lieutenant, Hellenic Navy
B.A., Hellenic Naval Academy, 1977

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 1988

Authors: ______________________
Antonios K. Makris

Approved by: ______________________
Vincent Y. Lum, Co-Advisor

____
John B. Isett, Co-Advisor

____
Robert B. McGhee, Chairman,
Department of Computer Science

____
Kneale T. Marshall,
Dean of Information and Policy Sciences

ABSTRACT

The Naval Officers Personnel Management System is a very complex system especially inside the Fleet Command. Managing the system manually is neither effective nor efficient in supporting the decision makers.

This thesis proposes a method to use a computer based information processing system to help decision makers in scheduling the assignment of officers to warships during the annual assignment process, as well as in other functions concerning personnel management. The thesis presents a decision support database system for the Naval Officers Management Staff.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

DEDICATION

To my wife [redacted]

To my son [redacted]

Who was born on [redacted] in my country of [redacted] during the third quarter of my studies in NPS.

ACKNOWLEDGMENTS

I would like to thank some of the people who contributed directly or indirectly to the publication of this thesis by providing essential help at some of the crucial turning points in my studies in NPS, as well as in my life.

My thesis advisors Dr. Vincent Y. Lum and Major John B. Isett, for their help to me by guiding me through this thesis and suggesting improvements and corrections, as well as for giving encouragement.

My medical advisor, Doctor William F. P. Tuer, COL, MC, Chief, department of medicine Allergy/Immunology Services in the Medical Department Activity Fort Ord, who gave me medication service as well as encouragement, with very effective result.

To the above people I am very grateful and I would like to express my deep gratitude.

I. INTRODUCTION

The introduction chapter sets the scene and prepares the reader for what is to come. This chapter identifies the background, the objectives scope and direction of the thesis, and the organization of chapters.

A. INTRODUCTION TO DATABASE

About 1970 a term appeared in the computer literature describing a new concept. The new term was "database." In the early 1970s database processing was considered an esoteric subject of interest only to the largest corporations with the largest computers. In recent years database processing has become an essential part of an organization's information system.

The information system supports the organization's functions, maintaining the data for these functions and assisting users to interpret the data for decision making. The database is an important tool in this process; it is not only the container of the data in the information system [Hawryszkiewycz, 1984:p. 1], but also a key module in the whole process. The management information system (MIS) intends to retrieve, extract and integrate data from various sources in order to provide timely information necessary for

Decision support systems (DSS) is one of the important applications of database. A Decision support system utilizes decision rules and models coupled with a comprehensive database and the decision maker's own insights, leading to specific, implementable decisions in solving that would not be amenable to management science optimization models [Turban, 1988:p. 73].

There are several issues related to the application of artificial intelligence (AI) in systems that automate or support problem-solving, diagnosis, advising, decision-making and control [Tanimoto, 1987:p. 461]. It is a technology of information processing concerned with processes of reasoning, learning, and perception.

Today, computers have become part of our life. They are common in industry, government, science, politics, and even in homes. As more and more organizations use computers, it is necessary to use systematic, new, and cost effective approaches for software solutions to their problems. One approach which is widely used in the computer world is the database system. Database systems play a central role in the computer world because of their facilities and data handling capabilities.

During the last decade the cost of labor has been increasing steadily in parallel with the increase of the software cost. Meanwhile the cost of computer hardware has decreased dramatically [Kroenke, 1983:p. 1].

Thus, simply stated, people have become more expensive as machines have become cheaper. The changing hardware over software cost ratio (H/S) has been steadily decreasing over time. In 1960 H/S the ratio was approximately 80/20. By 1980, the ratio was reversed in 20/80. By 1990 it will be about 10/90. These considerations lead us to select systems that achieve the best utilization of the software, and motivate system designers to build advanced database systems in order to decrease the software cost and obtain the maximum benefit [Fairley, 1985:p. 8].

Developing a database system allows system developers to meet the needs of the organization more effectively. The development process involves hundreds of discrete steps which are included in five general phases. These major phases of the development process are the following [Kroenke & Dolan, 1988:p. 75].

1. Definition Phase.
2. Requirements Phase.
3. Evaluation.
4. Design.
5. Implement.

During the first phase, the "definition phase", the problem is defined and the feasibility of a computer-based solution is examined. In phase two, the "requirements phase", the system requirements are specified in detail. During the third phase, the "evaluation phase", alternatives for meeting the users' needs are examined and one of the alternatives is selected. These first three phases can be combined into one, the called "analysis phase".

The next step of the project is to continue into the "design phase". It includes the description of files, data items, file relationships and the structure of the database (schemas, subschemas).

Once a design is complete, the final phase "implement phase" is developed. The details of implementation greatly depend on the particular database management system (DBMS). This phase involves coding and testing programs, converting data to the new system, and training personnel.

B. *BACKGROUND*

The most important resource in any organization is its people. Decisions and demands which affect personnel management have significant results, especially inside the military environment.

The military requirements for economic, medical, battle planning, personnel classification and organization, storage organization and work scheduling information are of primary importance to any defense organization. Generally today, information needed in an important personnel decision is available somewhere in the organization, but is not available to the decision makers when they need it.

The military personnel administration life cycle consists of six functions: Procurement, Education and Training, Assignment, Treatment, Promotion and Separation. Each function must be carefully planned and followed by a decision that requires information support.

It is necessary to adopt a more accurate, sophisticated and wide variety of information system. It is impossible to obtain all information needed through manual systems when the information is needed within a relatively short period of time. This thesis argues that a computerized database system can support personnel decision makers, specifically for the Hellenic navy.

The Hellenic Navy General Staff (HNGS) is organized into four major branches:

1. Fleet Command (FC).
2. Navy Logistic Command (NLC).
3. Navy Training Command (NTC).
4. Headquarter General Staff (HGS).

as shown in Figure 1.1.

This research will analyze the Fleet Command (FC). Currently, all information required by the director of the HNGS for scheduling and processing data about the officers of the FC is handled manually by his staff. HNGS needs accurate and timely information in order to make fast and better informed decisions.

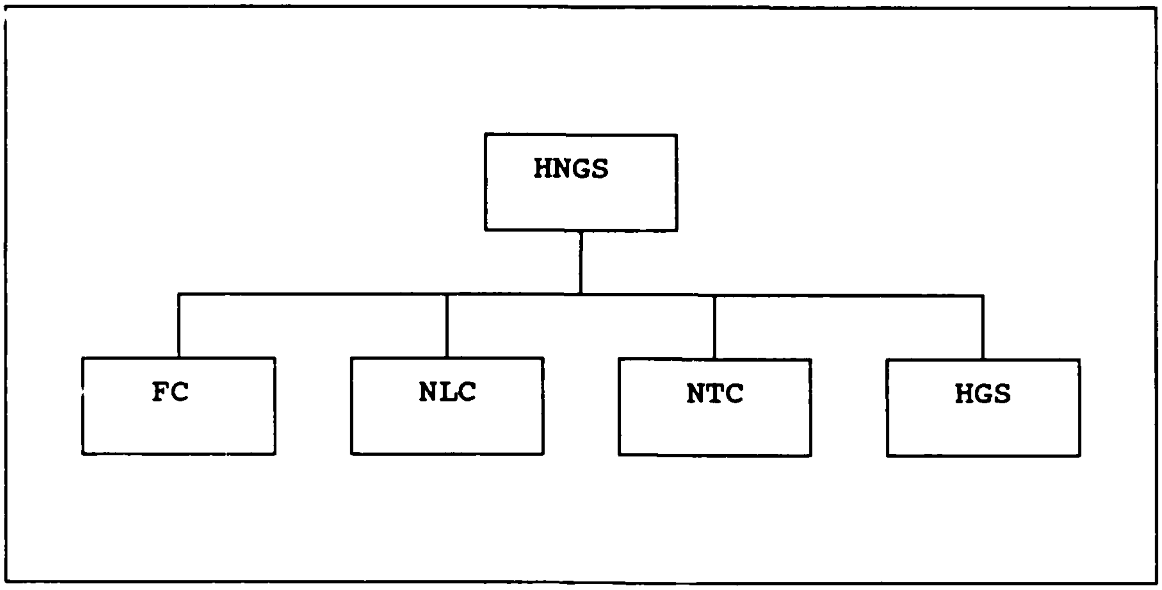


Figure 1.1 Basic Organization Chart of the HNGS

Generally, from this author's experience, it is possible to conclude that there are three main factors which limit human performances. These factors can be defined as:

1. Limited memory capacity.
2. Low execution speed.
3. Low accuracy factor.

The basic idea of our database for personnel management is to provide a means of extending all three of these factors. A microcomputer can support this idea of extension. But the best results are obtained by combining human memory and computer in cooperation. The human designs and gives orders, the computer constructs and executes.

Because of the complicated character of the job, especially inside the fleet, and the continuous changes concerning personnel and associated data, it is extremely difficult for the staff personnel to keep track of these changes and their results. Under these circumstances, the existing method of officer career management, is neither effective nor efficient in supporting the decision making process. Moreover, it is generally true that resources, especially personnel in the Hellenic Navy, are limited. Assignments to ships, for example, frequently run into many constraints which include among others, the appropriate rotation date, level of experience or skill, officers' requests, rank, and specialty.

These problems could be solved by computer support through the use of a microcomputer. This thesis will propose a method to use the computer to support the HNGS/FC personnel management process.

C. OBJECTIVES AND SCOPE OF THE THESIS

In light of the above, this thesis will develop the design for an automated personnel database system. It will discuss personnel management, collecting data relevant to personnel so that it can be used for a variety of applications. It will investigate the use of a microcomputer for processing personnel management data.

The development of a personnel database system has several advantages over the manual system. Fewer people are needed to do the same job, releasing manpower for other tasks and reducing the cost. At the same time it will reduce or eliminate data duplication, and this improves data accuracy or consistency. Information can be retrieved much faster and with a higher degree of accuracy. It will improve quality and speed of decisions. Lastly, it may provide better security.

The research will show how a database system can provide the appropriate, accurate information in a satisfactory and timely interval in order to assist the Deputy Chief Personnel and Staff Function under the Chief in decision making, regarding personnel management activities.

The action field of this research is within the Hellenic Navy Fleet (HNF), specifically the Fleet Command, where one finds the most complicated, and interesting tasks of the Hellenic Navy General Staff. The research will provide the framework of a system that will help decision makers to schedule and process the assignments of officers and to produce the most useful reports. Also, it will be able to track an officer's career, and to provide current information about an officer.

The developed system is considered a prototype which will need further implementation to be fully operational. Because of flexibility of the system, small further modification will provide a large area of application.

A model of personnel organization has been selected using the Hellenic navy standards. However, because of the unclassified nature of this research project, some details will be omitted. The system is "menu-driven", hiding details from the user and providing him with a friendly environment.

The Relational Data Model (RDM) is selected as the appropriate model. The dBASE III PLUS software package is used as an example of Database Management System (DBMS) because it includes both a data manipulation language and a general purpose programming language, offering a host of ways to manage information.

D. ORGANIZATION OF CHAPTERS

Chapter II reviews the basic concepts of a database processing system. It includes some basic definitions, the architecture of a DBMS, the file and database processing system, the advantages and disadvantages of database processing, and the basic database models with their comparison. It also presents an overview of relational database design and microcomputer database.

Chapter III describes the system analysis and requirements. It presents the Naval personnel administration functions, the current system with its existing problems, the assignment mechanism and criteria, the system goals and requirements, and the system input and output information.

Chapter IV describes the system design. It contains the logical and physical design. The logical design presents the classes of system functions, the entity-relationship process, the relational database scheme, the relation definition and classification, and the database dictionary. The physical design presents the DBMS specifications and the hardware specifications.

Chapter V presents the assignment model development and the system implementation. It describes the assignment model design and its strategy, and demonstrates the system implementation through the menu-driven control mechanism.

Chapter VI presents the conclusions and recommendations based on this research, and gives the future development of the system and its possible extension.

II. GENERAL DATABASE CONCEPTS

In this section some definitions and basic database terminology are provided, followed by a summary of database architecture, types of data models, relational database design, and microcomputers. These are the most important concepts that compose the base of building this research.

A. DEFINITION AND BASIC TERMINOLOGY

While reviewing this research the reader will find much terminology related to the database. In order to make sense, the meaning of the most common and useful of this terminology is explained.

A starting point is the "database," which is a shared collection of interrelated data designed to meet the varied information needs of an organization. Closely related to the database is the "database management system (DBMS)." This is a software system that performs all user requests for data (insert, delete, update, retrieve). A presupposition of these is the existence of the "database system," that is, a system to record and maintain information that is significant to an organization in the decision making process. It is also called information system. A portion of a database, named "Application," is a collection of menus, reports, and programs that addresses the needs of a user group.

The interface of a DBMS has two main components, the "data definition language (DDL)" and the "data manipulation language (DML)." The DDL is a specialized language used for the description of the database (records and data-items). This description is stored in the data dictionary maintained by the DBMS. The DML is a programing language used to formulate queries or to write application programs for data manipulation. It is also called the query language.

The unit of description of the world, called "entity," is an object that exists and is distinguishable from other objects. An entity is represented by a set of attributes. An "attribute" or "field" is a property of an entity, the smallest unit of named data. The set of permitted values of an attribute is called "domain." "Entity set" is a set of entities of the same type.

The "file" that is used for the data is an organized collection of records representing entities of the same type. The "record" is a collection of data representing one entity of a file. A file generally has in its attributes a "key," that is, an attribute or a set of attributes, whose value uniquely identifies each entity in a file.

The existence of the entity sets usually implies the existence of associations among them. An association represents a "relationship" between entity sets (or files) and is an ordered list of these entity sets. Binary relationships are classified into the following four categories according to how many entities from one entity set can be associated with how many entities of another entity set [Korth & Silberschatz, 1986:p. 25]. Figure 2.1 illustrates the four categories of relationships. Let A, B be entity sets. The relationship between A and B must be one of the following:

1. One-to-one relationship (1:1). Each entity in A is associated with at most (exactly) one entity in B, and each entity in B is associated with at most one entity in A. Figure 2.1a.

2. One-to-many relationship (1:N). Each entity in A is associated with any number (many) of entities in B. Each entity in B, can be associated with at most one entity in A. Figure 2.1b

3. Many-to-one relationship (M:1). For each entity in A there is at most one associated entity in B. For each entity in B, however, there exist any number of associated entities in A. Figure 2.1c.

4. Many-to-many relationship (M:N). Each entity in A is associated with any number of entities in B and each entity in B is associated with any number of entities in A. Figure 2.1d.

B. ARCHITECTURE OF A DATABASE SYSTEM

It should be obvious that between the computer, dealing with bits, and the ultimate user dealing with abstractions such as military units or assignment of personnel to a division, there are many levels of abstraction [Ullman, 1982:p. 6].

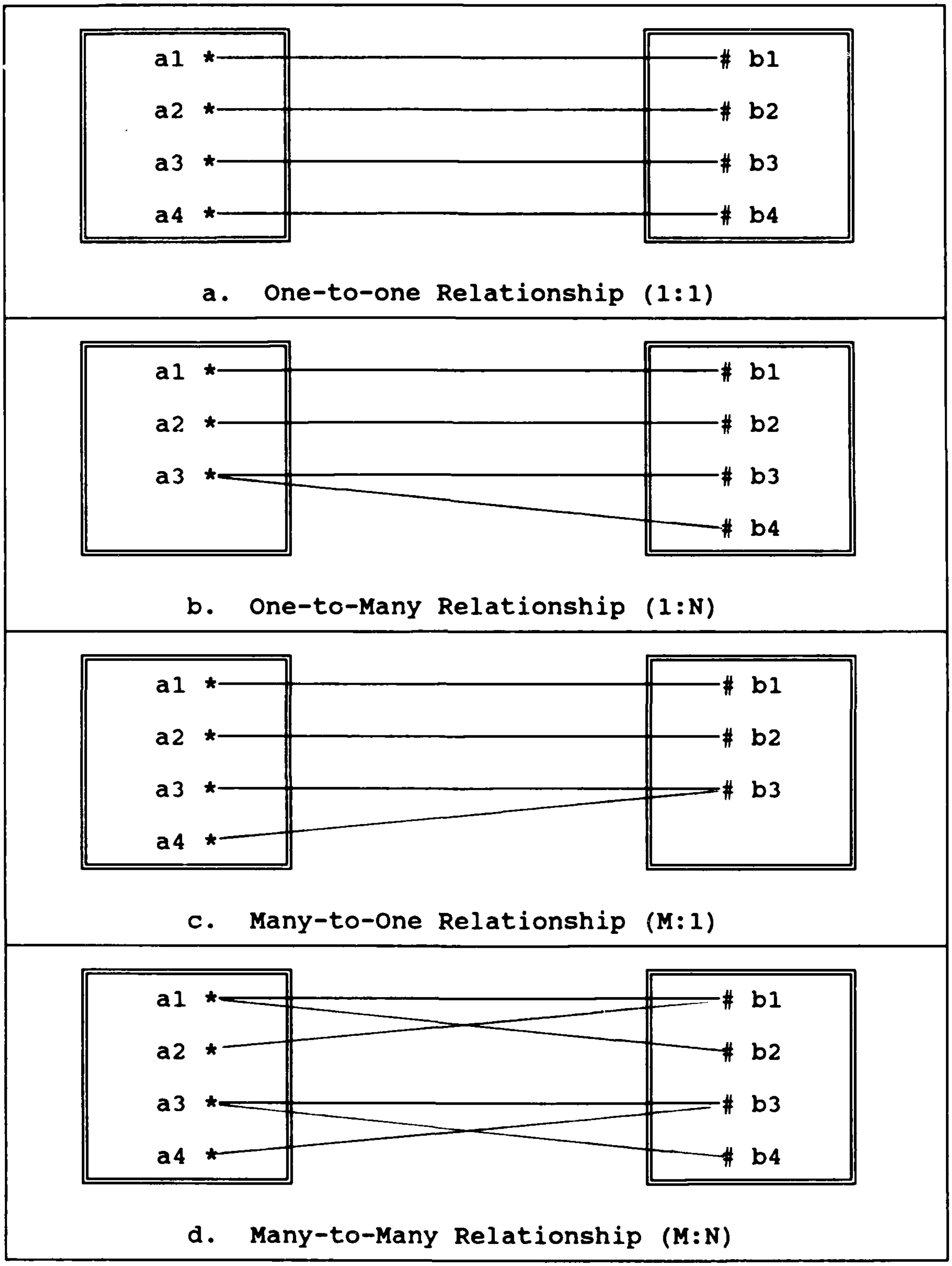


Figure 2.1 Categories of Relationships

The complexity of the system is hidden from the users. A major purpose of a database system is to provide users with an abstract view of the data. Each view in a database architecture represents a level of abstraction [Korth & Silberschatz, 1986:p. 4].

The database architecture is divided into three different levels of abstraction. The "internal" view, the "conceptual" view, and the "external" view. Figure 2.2 illustrates the standard view-points regarding the three-level of a database architecture [Howe, 1983:p. 26].

The "internal" view is the physical view of database and resides permanently on secondary storage devices such as disks and tapes. This is the lowest level of abstraction, at which one describes how data are physically arranged and how they are allocated into files.

The "conceptual" view, also called "schema," is the complete, logical view of data. That is, the data are represented in such a way that can be understood by a human. This is the next higher level of abstraction which describes what data actually stored in the database and the relationships that exist among data. A DBMS provides a data definition language (DDL) to specify the conceptual view.

The "external" view or "subschema" is the highest level of abstraction. It describes only part of the conceptual view, representing an application of the entire database. It also called user view.

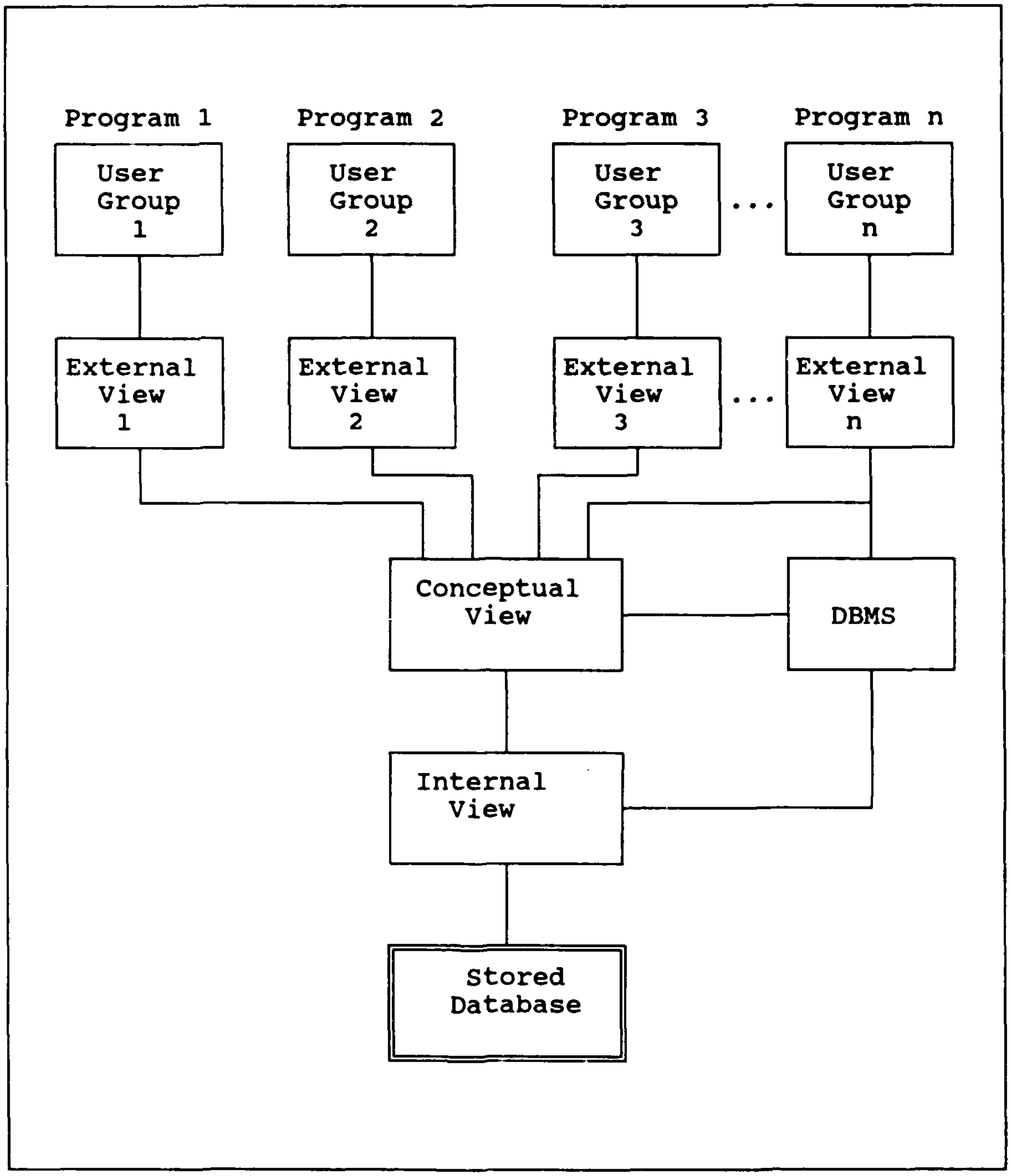


Figure 2.2 Levels of Abstraction in a Database System

[Howe, 1983:p. 26]

A user can interface with a database either via an application program or via an interactive query language. Each external view is specified using the data definition language (DDL) and each application program uses the data manipulation language (DML) to access the database. All of the retrieval and update actions expressed via the data manipulation language (DML) are executed under the control of the database management system (DBMS).

C. FILE AND DATABASE PROCESSING SYSTEM

Database technology allows an organization's data to be processed as an integrated whole. It reduces the artificiality imposed by separate files for separate applications and permits users to access data more naturally [Kroenke, 1983:p. 1].

Figure 2.3 shows three traditional file processing systems [Kroenke, 1983:p. 2]. Each file is considered to exist independently and each system processes its own file, resulting in no sharing of data within a system.

Figure 2.4 shows a database processing system [Kroenke, 1983:p. 4]. The file systems have been integrated into a single repository called "database," which is processed indirectly by the application programs. This system can perform all the functions, but the programs call the Database Management System (DBMS) to access the database. The DBMS is the bridge between application programs and the data base.

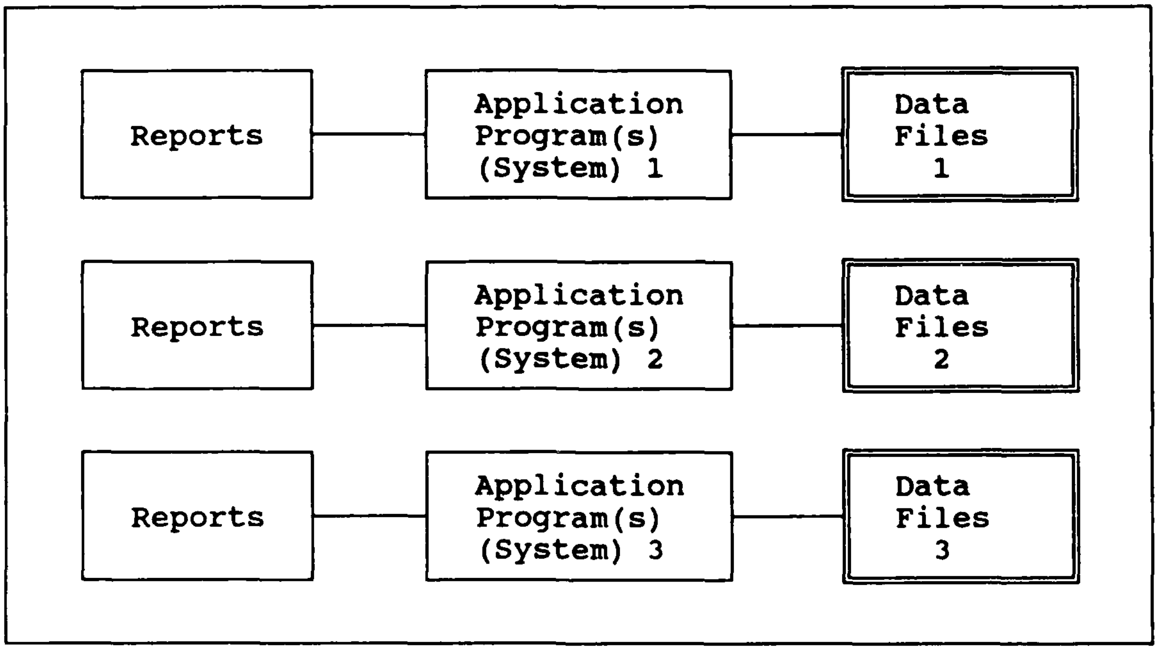


Figure 2.3 File Processing System [Kroenke, 1983:p. 2]

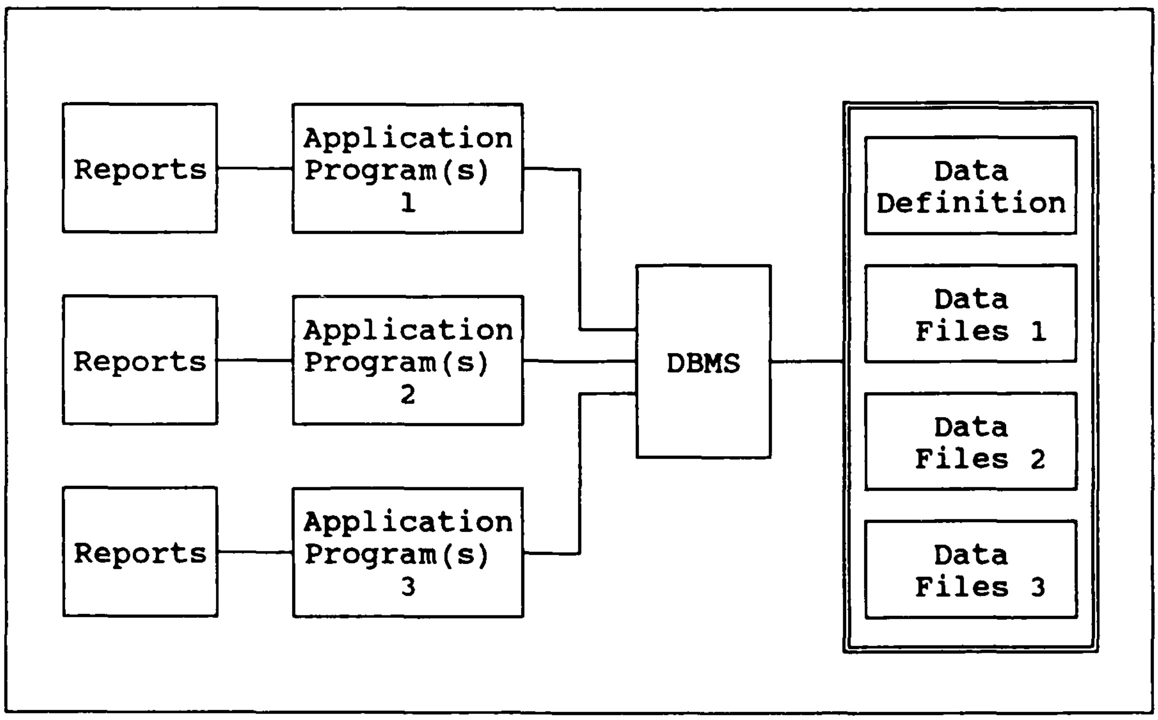


Figure 2.4 Database Processing System [Kroenke, 1983:p. 4]

It is a complex and usually large program that acts as a data librarian. In order for the DBMS to perform its functions, it stores not only data, but also a description of the format of the data [Kroenke, 1983:p. 3].

D. ADVANTAGES AND DISADVANTAGES OF DATABASE PROCESSING

Database processing has its own advantages and disadvantages over file processing [Kroenke, 1983:p. 3]. These are:

1. Advantages

a. Enables the users to derive more information from a given amount of stored data by integrating the relationships into a database. This is because DBMS allows processing of any combination of data stored in the database, and thus we can obtain more information.

b. Eliminates or reduces the data duplication, and so minimizes data redundancy. A data item may be recorded once, while in the file processing system the same information can be repeated in different files. Elimination of duplication saves file space and to some extent, can reduce processing requirements. The most serious problem of data duplication is that it can lead to lack of data consistency or integrity.

c. Supports program and data independence. When data structure changes, application programs keep running without being changed. DBMS isolates any change in file formats, record structure, etc. from application programs. In the file processing system, the structure of files is distributed across the programs creating problems when a file is changed.

d. Provides data consistency. There is a less chance of inconsistency because the redundancy is controlled.

e. Allows sharing of data. Data can be shared by many application programs through the DBMS. In the file processing system, since every application has its own private files, there is little opportunity to share data from other application program files.

f. Provides better data management. When data is centralized in a database, one department can specialize in the maintenance of data. Furthermore, centralization of data management leads to economical gains. One person working full time on data problems can be more efficient than 20 people working one-twentieth of their time on the same problems.

g. Allows the users to interface with the query languages for easier programming and application development.

2. Disadvantages

a. Can be expensive. The DBMS may occupy so much main memory that additional memory must be purchased. Conversion from existing systems can be costly, especially if new data must be acquired. Once the database is implemented, operating cost for some systems will be higher.

b. Tends to be complex. Large amounts of data in many different formats can be interrelated in the database. This structure means more sophisticated programming. Application system design may take longer, and of course highly qualified systems and programming personnel are required.

c. Tends to be difficult for backup and recovery. This is because of increased complexity and because database are often processed by several users concurrently.

d. Increases vulnerability to security problems because all data are centralized under one system.

E. DATA MODELS

The study of database processing involves learning the database models. In this section the basic data models are presented.

A model is a representation of real word objects, events, and their associations in a mathematical form. A data model is an abstract representation of the data about entities, events, activities, and their associations. The purpose of a data model is to represent data in an understandable way. In general, a data model consists of two elements [Ullman, 1982:p. 18]: First a mathematical notation for expressing data and relationships, and second operations on the data that serve to express queries and other manipulations of the data.

Three kinds of data model are most important today:

1. Hierarchical data model.
2. Network data model.
3. Relational data model.

1. Hierarchical Data Model (HDM)

A hierarchical data model consists of a collection of records (nodes) which are connected with each other through links. Each record is a collection of fields (attributes) each of which contains only one data value. A link is an association between precisely two records.

The tree-structure diagram is the scheme. Thus the hierarchical database consists of one or more trees and each tree consists of a hierarchy of records. The link represents one-to-one or one-to-many relationships from a parent node to its child node. Figure 2.5 shows the hierarchical data model [Yao, 1985:p. 84].

The basic operation on a hierarchical database is a tree walk, that is, given a node of the database instance we can search all of the descendants of this node.

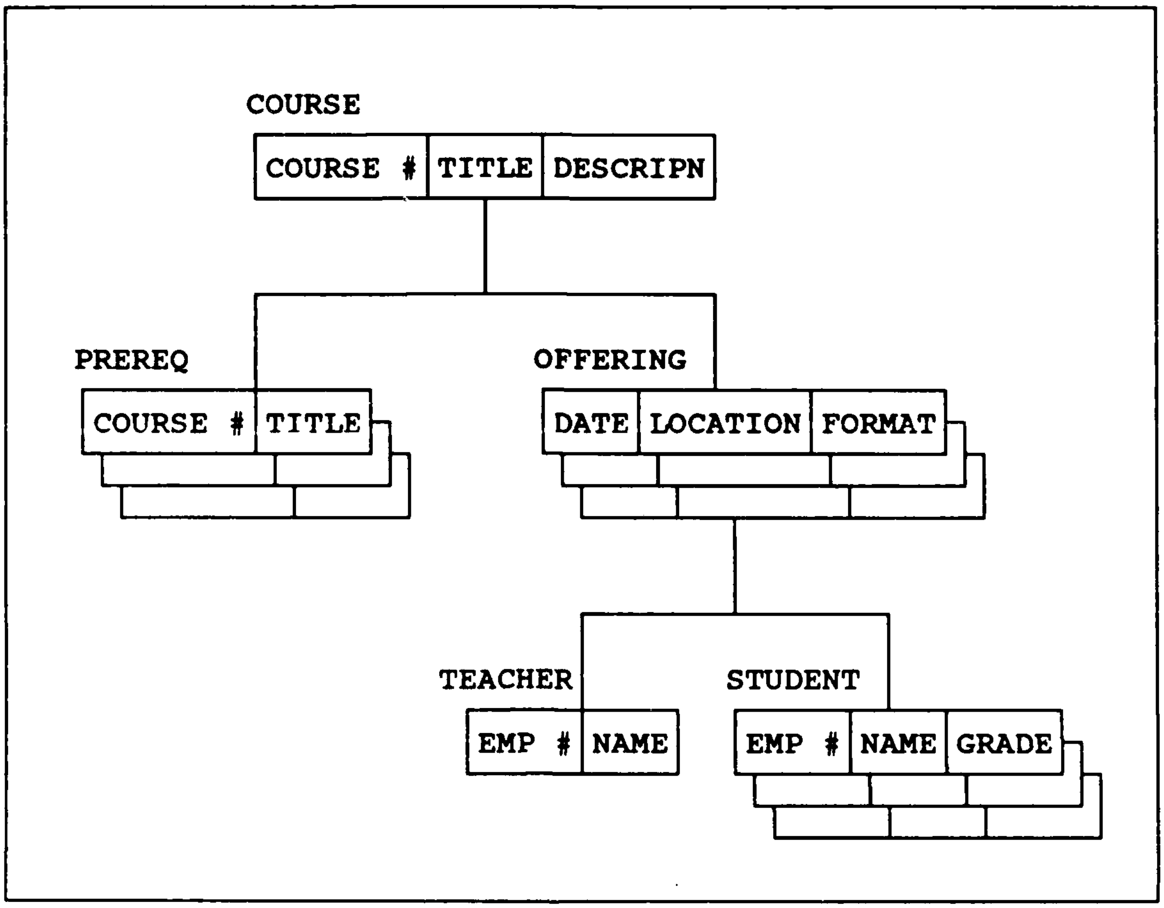


Figure 2.5 Hierarchical Data Model

[Yao, 1985:p. 84]

The advantage of this data model is that the tree structure is well known and widely used. The disadvantage is that this model cannot easily support the many-to-many and many-to-one relationships.

2. **Network Data Model (NDM)**

The network data model is similar to the hierarchical model in the sense that data and relationships among data are also represented by records (nodes) and links, respectively.

The basic data structure used in a network database is the graph. The links in the graph are bidirectional. Thus the hierarchical model differs from the network model in that the records are organized as collection of trees rather than arbitrary graphs. Figure 2.6 shows a representation of a network data model [Kroenke & Dolan, 1988:p. 186].

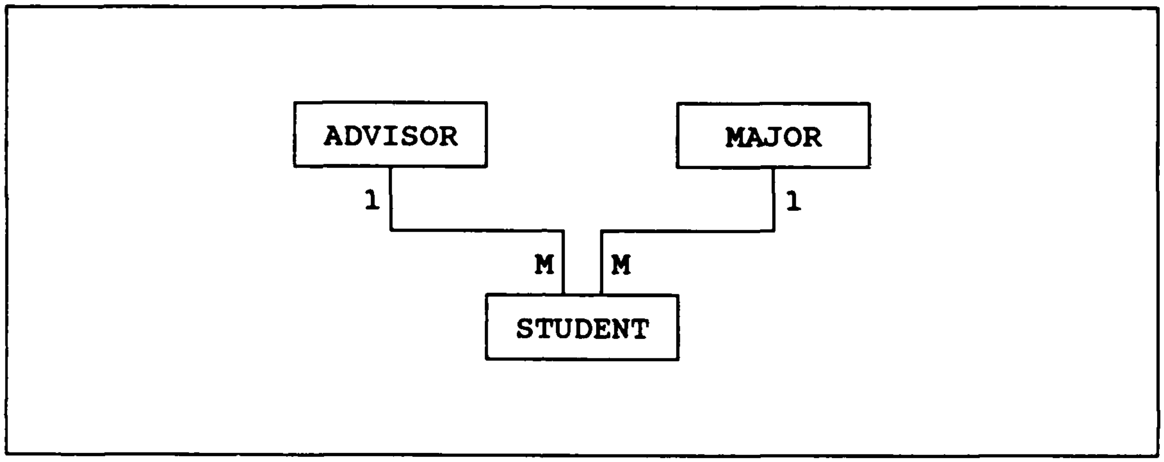


Figure 2.6 Network Data Model

[Kroenke & Dolan, 1988:p. 186]

The links in the graph represent multiple one-to-many and many-to-one relationships between the same pair of record type. Relations that involve more than two record types are not directly permitted. A node child is called member in network terminology and a node parent is called owner.

The network data model can be considered as an extension of the hierarchical data model since the tree structure is a special case of graph.

The major limitation of this data model is its inability to express many-to-many relationships simply.

3. Relational Data Model (RDM)

The relational data model differs from the hierarchical and network data model. It consists of a collection of tables (relations) and there are no links and nodes. There is a direct correspondence between the concept of a table and the mathematical concept of a relation. We introduce some set-relation theory definitions in order to understand this model better [Ullman, 1982:p. 19].

a. A set is a collection of well defined objects which are called elements or members of the set.

b. A domain Di is simply a set of values.

c. A relation is any subset of the cartesian product of one or more domains (list of domains).

d. Tuples are the members of a relation.

A relation is a two-dimension table with a unique table name. The table name is also termed the name of the relation (or briefly, relation name). Every table is made of columns and rows. Each row, except the first one, is a tuple and represents the file record. Each column has a distinct name, the name of the attribute, and contains values about that attribute (field). The column headings are attribute names. The number of columns of a table is the degree of the relation. The number of rows (not counting the column heading) i.e. the number of tuples of a relation is termed the cardinality of the relation. Figure 2.7 represents a relational data model.

PARTS

PARTNO	PARTNAME	COST
1001	KEYBOARD	175.80
1002	DRIVE	192.55
1003	CHIP MEM.	6.15
1004	PRINTER	355.60
1005	MONITOR	280.30
1006	MODEM	95.75
1007	DISKETTE	1.05
1008	POWER SU.	128.32

ORDERS

CUSTNO	CUSTNAME	PARTNO	QUAN
0001	SMITH	1003	9
0002	JONSON	1005	2
0003	MILLS	1002	2
0004	ROBERTS	1001	1
0005	WILSON	1007	50

Figure 2.7 Relational Data Model

The set of attribute names for a relation is called the relation scheme. The collection of relation schemes used to represent information is called a (relational) database scheme, and the current instantiation of the corresponding relations is called the (relational) database.

Using the table analogy, the two-dimension table representing relation must satisfy the following conditions:

a. Each column contains values about the same attribute.

b. Each column has a distinct name.

c. Each row is distinct.

d. The sequence of the rows is immaterial.

The principal advantage of a RDM is that it supports all types of relationships, that is, one-to-one, one-to-many, many-to-one, many-to-many. Also tables are more understandable than the graphs and trees. Thus the way of arranging the data is simpler and more understandable for humans.

For the above reasons, even if it is the newest model (introduced in 1970 by E.F. Codd), it has become the most popular one.

F. COMPARISON OF DATABASE MODELS

To evaluate the three models and to select one among them, there are two main standard criteria by which they should be judged in order to achieve the objectives of a database system organization [Ullman, 1982:p. 168].

"Ease of use": It requires less time for users to become familiar with the database system. The principal cost may be time spent by the programmer writing applications programs and by the user posing queries. We want a model that makes accurate programming and the phrasing of queries easy.

"Efficiency of implementation": For large database the cost of storage space and computer time (execution time) spent dominate the total cost of implementing a database.

By the criterion of easy of use, the relational model is superior than the others. It provides only one concept, the relation (table), that the programmer or user must understand. Moreover, the relational algebra and calculus clearly provide a notation that is quite powerful. This model, also, adopts very high level languages for expressing queries concerning data represented. The network model as graph-structure, requires understanding of both records types and links, and their interrelationships. The implementation of many-to-many relationships and relationships on three or more entity sets is complex and not straight forward. Similarly, the hierarchical model with its tree-structure and requires understanding the use of pointers (virtual record types). It also has the same problem as the network model regarding the representation of relationships that are more complex than many-to-one relationships between two entity sets [Ullman, 1983:p. 169].

The relational data model (RDM) supports all types of relationships (one-to-one, one-to-many, many-to-one, many-to-many). It makes no distinction between the representation of relationships and of entities. Both are supported as relations. Therefore relationships, like entities may have attributes specified and must be stored in the data.

The relation DBMS most naturally process data in the manner of an entire file at a time and can be used for most applications. But one application type has found the relational model to be particularly useful, namely decision support systems (DSS). Because the products based on the relation model are generally easy to use, relational database is often the heart of DSS [Kroenke & Dolan, 1988:p. 23].

In the standard of efficient implementation, the network and hierarchical models seems to win. Since relations can, and often do, represent many-to-many relationships, the relational models balances this criterion by adding an intersection relation with the fields including at least the keys of the two relations.

Early commercial database systems were almost uniformly based on the network or hierarchical model, because the emphasis of such systems has been on the maintenance of large databases, and these models lend themselves most easily to the necessary efficient implementation. However, since 1982 there were several successful commercializations of the relational model, and Ullman found that the relational systems will become progressively more accepted for two reasons: First it is becoming clear that the same concepts used to design a large database apply as well to small and medium scale databases, and there are many more small databases than large ones. With small databases, the ease of use inherent in the relational model assumes increased importance. Second, many of the apparent inefficiencies of the relational model can be eliminated [Ullman, 1982:p. 170].

Through the above discussion, the relational model is considered better than others for the Hellenic navy officers (HNO), since this system is easy of use and the whole officer manpower is not large, fluctuating around 2500 officers. Also the users have little knowledge of database systems and languages. Therefore, they require a database system that does not need great skills. Then, the potential of efficiency in the relational model can be increased using the normalization process. In these situations the relational model is more helpful than others.

G. RELATIONAL DATABASE DESIGN

In general, the goal of a relational database design is to generate a set of relation schemes that allow us to store information without unnecessary redundancy, yet allcw us to retrieve information easily. One approach is to design schemes that are in an appropriate "normal form". In order to determine this normal form, we use additional information, a collection of constraints called "data dependence" [Korth & Silberschatz, 1986:p. 173].

The normal forms defined in relational database theory represent guidelines for record design. The normalization rules are designed to prevent update anomalies and data inconsistencies.

1. Functional Dependency (FD)

Functional dependencies are constraints of the set of legal relations. They allow us to express facts about the enterprise that we are modeling with our database [Korth & Silberschatz, 1986:p. 181]. More simply a functional dependency is a relationship between attributes. A set of attributes Y is said to be functionally dependent on a set of attributes X, if the value of X functionally determines the value of Y.

In term of mathematical theory let R be a relation scheme and X, Y are subsets of it [Korth & Silberschatz, 1988:p. 182]. The functional dependency X --> Y holds on R if in any legal relation r(R), for all pairs of tuples t1 and t2 in r, t1[X] = t2[X] implies that t1[Y] = t2[Y]. The set of attributes X is known as the determinant of the functional dependency X --> Y.

Using the functional dependency concept, K is defined as a superkey of R if K --> R. That is, K is a superkey if whenever t1[K] = t2[K], then t1[R] = t2[R] (that is t1 = t2). In general a key is an attribute or set of attributes that functionally determines the non-key attributes.

Every relation has at least one key which uniquely identifies a tuple of this relation.

Functional dependencies are important because they lead to several highly desirable normal forms for relational database. In order to find keys, we need to determine all the functional dependencies that hold.

2. Normal Forms (NF)

Some relations, although they contain usable data, can have undesirable consequences when updated by changing these data. This phenomenon is called modification anomalies. Identifying and eliminating modification anomalies is the essence of the normalization process. Normalization is the process by which attributes (data items or properties) are grouped together to form new relations. The classes of relations obtained through the techniques for preventing anomalies are called normal forms.

There is a series of normal forms which describe the relationships of data within the relational tables. Figure 2.8 gives the relationship of all normal forms which is most useful in understanding the different levels [Kroenke & Dolan, 1988:p. 137].

The focus of the process is to develop tabular relations that are the most complete, logical and redundant-free. Depending on its structure, a relation of the proposed system might be in first normal form, second normal form, third normal form, or Boyce-Codd normal form.

A relation scheme is in first normal form (1NF) if the domains of all attributes are atomic. A domain is atomic if elements of the domain are considered to be indivisible units. Under first normal form, all tuples in a relation must have the same number of attributes. There are no repeating groups of the data items within the tuple (record). Every normalized relation is in the first normal form.

A relation is said to be in second normal form (2NF) if it is in first normal form and every non-key attribute of the relation is fully functionally dependent on the primary key. That is, if all nonkey attributes are dependent upon all attributes of the key.

A relation is in third normal form (3NF) if it is in second normal form and has no transitive dependencies. All attributes must depend upon all of the key and dependencies are not transferred from one attribute to another.

A relation is in Boyce-Codd normal form (BCNF) if every determinant is a candidate key.

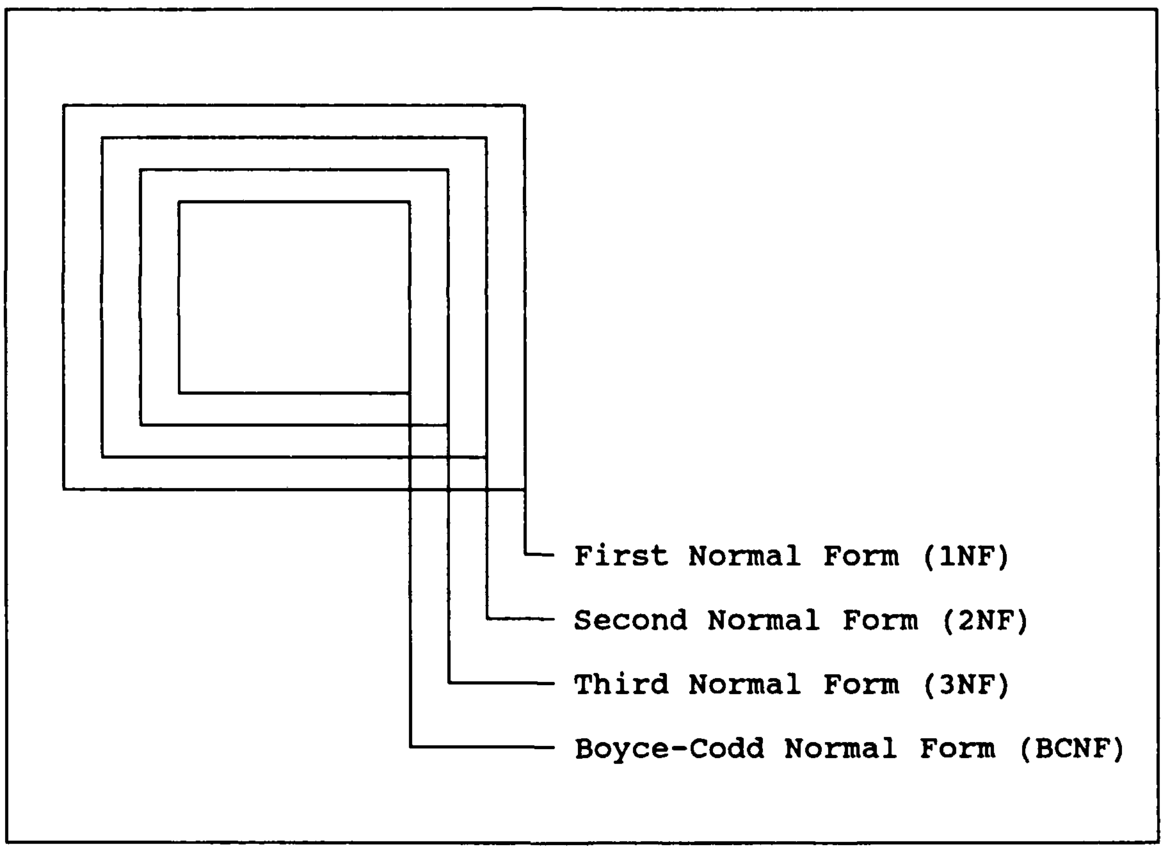


Figure 2.8 Relationship of Normal Forms

[Kroenke & Dolan, 1988:p. 137]

H. MICROCOMPUTER DATABASE

1. The Microcomputer Environment

The advent of 16-bit microcomputer brought database processing to the masses because database processing became cost-effective and affordable. Admittedly, microcomputer databases are small when compared to large mainframe database. But the principles, the development process, and the needs for administration are nonetheless the same. [Kroenke & Dolan 1988 :p. 341]

Microcomputers are a recent technology and are accepted by many people because they provide several advantages as compared to the large computers. First, microcomputers are not as expensive as the mainframe, and can be helpful for many people for personal use. Second, they are powerful, reliable, and can be used for a wider range of specific applications. Third, microcomputers are acceptable in any environment and can replace the older computers, which require additional funding for maintenance personnel and special facilities (air conditioned rooms and large spaces).

However, the major disadvantages of microcomputers are the limited access speed and the limited storage.

At about the same time relational DBMS were becoming widely used in decision-support systems, making access to the corporate database easier for users, the microcomputer explosion occurred, making computer power even more available. The combination of microcomputers and relational data model presented some tremendous opportunities in end-user database processing [Kroenke & Dolan, 1988:p. 23].

The primary characteristic of microcomputer database systems is simplicity. The limited capability of personal computers limits both the size of the database and the degree of the sophistication of the system. As the power of personal computers has grown, so has the sophistication and complexity of microcomputer database system.

2. Classes of Microcomputer Database

There are three fundamentally different kinds of microcomputer database [Kroenke & Dolan, 1988:p. 344].

a. Stand-alone database (type I)

b. Imported-data database (type II)

c. Multi-user database (type III)

Figure 3.9 summarizes the three types of the microcomputer database.

a. Type I

Type I microcomputer database stand alone (Figure 3.9a). They neither receive data from nor send data to other microcomputer or terminals. They are usually employed by one or at most a few users. As a result they present few problems.

b. Type II

Type II microcomputer databases consist at least partly of data that is down-loaded, or imported, from another computer, usually a corporate mainframe (Figure 3.9b). Once

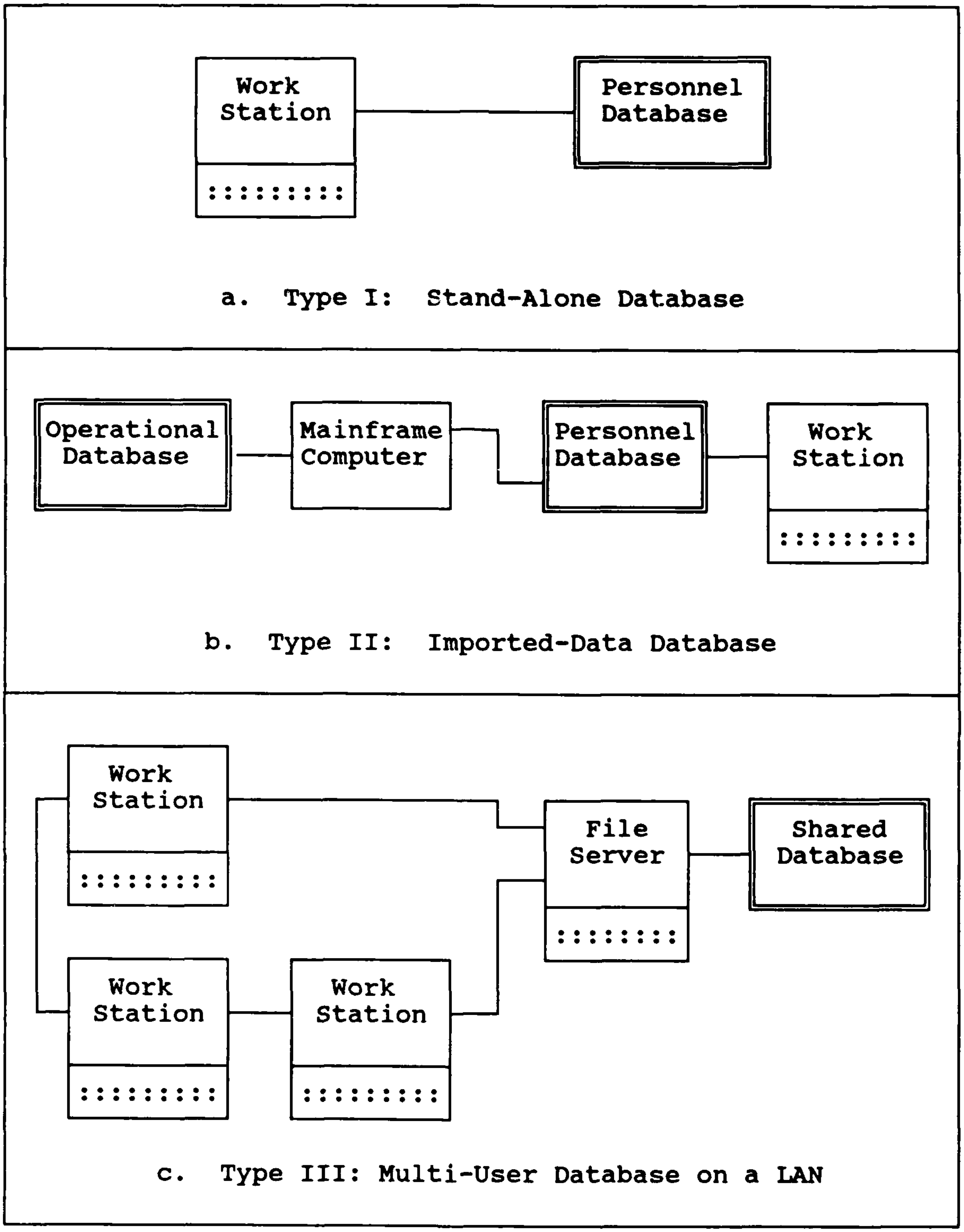


Figure 2.9 Types of Microcomputer Database

[Kroenke & Dolan, 1988:p. 346]

the data is stored on the microcomputer, it can be processed as if it were type I (stand-alone) database. The number of type II microcomputer database has increased dramatically in recent years. Many organizations have micro-to-mainframe links that allow microcomputers to communicate directly with the corporate mainframe.

Type II database that gets its data from another computer presents several problems, and requires special attention to coordination of activities on the microcomputer: data consistency, access control, and prevention of criminal activity.

c. Type III

Type III microcomputer database supports the concurrent updating of a shared database (Figure 3.9c). Most type III databases reside on a local area network (LAN). The common database is stored on one of the LAN's micros, called the file server. The microcomputers that want to process the database send requests for data actions to the file server.

Multi-user microcomputer databases are subject to concurrent processing problems because many microcomputer DBMS products do not yet contain the locking, recovery, or security features that are needed.

I. SUMMARY

Since the area of this thesis deals with the development of an effective database system, a necessary consideration of the author is to bring together a collection of state-of-the-art methods that address the theory that builds this research.

This chapter has provided the reader with the necessary background, presenting a review of the most essential steps, and some basic selections under comparisons, on which the research is based. The most useful terminology and the three levels of abstraction are discussed in order to give an explanatory connection between the computer and the user, because the complexity of the system is hidden.

The introduction of database processing and the discussion of its nature, advantages and disadvantages, gives the understanding of the reasons why this thesis will try to switch the existing manual system to a computerized database system.

For the selection process of a database model, the three principal models, network, hierarchical and relational are introduced and compared. The comparison shows that in this research the relational model has many advantages over the other two, and so this model is preferable.

As consequence of the relational model, the normalization techniques to the relational data structure are provided. The normalization is the judge of the relational design and establishes the rules under which relations are constructed.

Finally, since this thesis seeks a cost effective solution by computer support through the use of a microcomputer, the three fundamental classes of microcomputer database in which this research could take place are described.

III. SYSTEM ANALYSIS AND REQUIREMENTS

Systems development can generally be thought of as having two major components, system analysis and system design. System analysis is the process of gathering and interpreting facts, diagnosing problems, and using the facts to improve the system through better procedures and methods. System design is the process of planning a new system to replace or complement the old. In other words, analysis specifies what the system should do, and design states how to accomplish the objectives [Senn, 1984:p. 5].

A. NAVAL PERSONNEL ADMINISTRATION FUNCTIONS

There are six functions that constitute the naval personnel administration life cycle: procurement, education and training, assignment, treatment, promotion, and separation.

1. Procurement

Personnel procurement deals with the process of gaining new manpower from national human resources, for filling vacant positions according to the Hellenic Navy requirements. Data relevant to the candidates that have been selected, must be kept and maintained. Thus the staff selects data from the Naval Officers Academy relevant to the their career, so that these data can be used at any time for new assignment, promotions, etc. Among these data are, the nomination (graduation) date, and the order in class. The order in class can change during the officer's career because it is related to the annual schools.

2. **Education and Training**

Information relative to the personnel education and training function is used mainly for personnel development related to assignments and promotions. A person's educational background is used to gain special knowledge needed to place a person in a particular job and to prepare that person for a new assignment. Further, this information is used to plan and monitor the careers of leaders, or those with special abilities who may be future leaders.

The results of personnel development can be measured by observing the performance of individuals in gaining necessary skills and abilities. This information can be recorded in the personnel data and used as a basis for further career development.

3. **Assignment**

Personnel assignment is the function that deals with selecting the right people (officers) for the right positions. Three general aspects must be considered for this function.

First, every vacant position must be filled by a person with the ability to carry out the job in the best manner.

Second, the rank, specialty, and abilities of each person must be fitted to the job so that he satisfies the job area.

Third, some positions require that the selecting person must have finished compulsory education.

The assignments function is performed after executing the promotions function. This is the most complex and difficult job of the decision makers, who compose the assignment scheduling committee (ASC).

Since this research is focused in the assignment process, the criteria that affect the assignments of the officers in the warships will be examined and analyzed in separate section.

4. **Treatment**

Personnel treatment deals with the physical and psychological aspects of the person and job. These include such areas as mental and physical health, recreation, rewards, transportation, salary, retirement plans, insurance, vacation, pension, and allowances, (wife, children) etc.

Mental and physical health conditions and reward affect the promotion and assignment functions. Salary, military insurance, pension, and personnel service affect the life of the family.

5. **Promotion**

Officers promotion deals with the officers who have finished minimum service duration in a rank and possess the ability to perform in upper level positions. This is the job of the decision makers, namely promotion selection committee (PSC). The necessary information should be prepared and provided to the committee. The list of officers who can be promoted or not, should be provided according to rank, unit of service, and specialty. Also, the promotion point tables of all officers should be provided.

The most important factors that affect the promotions are: The standard requirements on the current rank, the reports about the officer, the education, and the physical and mental condition. The promotion selection committee (PSC) selects the officers to be promoted, and the process occurs normally in the period of May and June. There are lists of officers for each rank who are recommended for promotion according to the above information.

6. Separation

Personnel separation occurs when an officer voluntarily asks to be released from the navy or goes through the process of retirement or through the firing process. An officer can be fired for many reasons. Officers who request retirement must have worked for a minimum public service duration in the navy. If an officer reaches the age limitation, rank limitation or maximum public service duration, they must retire on that day. Therefore retirement information should be prepared and provided to decision makers. This information must include a list of officers who wish to retire and have satisfied the minimum requirements and the public service duration for each officer.

B. PROBLEM DEFINITION

As noted the Hellenic Naval Officers Personnel System is a manual system. The conclusion from the above discussion, about the aspects of the officer management, is that the system is also very complex. Because of the continuous changes concerning personnel and associated data, it is extremely difficult for staff personnel to keep track of these changes and their results.

Managing this system manually demands great efforts. It is an inefficient, tedious, time-consuming operation. Also, the chief may not be able to make fast decisions concerning officer management due to the lack of timely and accurate information. Furthermore the volume of transactions pertaining to officer management is getting larger and larger, which means that additional personnel are required to perform the above job, requiring more space, and increasing the complexity of the system.

One of the major responsibilities which also is the biggest problem for the decision makers, is to schedule and process the annual assignments of the officers inside the Fleet Command, that is, to perform the assignments of the officers to warships. Each officer during his career has to be assigned to various units according to some existing criteria. For this reason there exists a mechanism through which the staff schedules the assignments of its officers after each annual promotion process.

Although there are organization tables, general rules for assignments, and records of the characteristics (rank, specialty, education, etc.) of each officer, the assignment of officers to warships is a very complex and difficult task. This task is even more complex due to the existing requirements and constraints among officers, among ships, and between officers and ships.

C. DESCRIPTION OF THE CURRENT SYSTEM

The description of the current system is important because it is considered a basic step in order for the problem solution to be applicable and effective. Two major organization components compose and activate the assignment process inside the Fleet Command. These two components are the organization of the Fleet Command units and the organization of the Fleet Command officers.

1. Organization of the Fleet Command Units

As noted in Chapter I, the Hellenic Navy General Staff (HNGS) is organized into four major branches: Fleet Command, Navy Logistic Command, Navy Training Command and Headquarter General Staff. The Commander of the Fleet has under his flag all combatant ships. The Navy Logistic Command is responsible for the bases, the supply center and all auxiliary ships. The Navy Training Command is in charge of the Naval Officers Academy, Petty Officers School, training centers and training ships. Each command is divided into subordinate commands.

This research will analyze the Fleet Command (FC). The FC is organized into subordinate, staffs, warships. Figure 3.1 illustrates the basic organization chart of the Fleet Command, providing a summary of the relation among its units. As stated above schools and training centers belong to the Navy Training Command. This situation is referred to those kinds of schools that belong to the Fleet Command and not the assignments, and to those training centers that are considered warships. Schools that affect the annual assignments will be discussed in the criteria section.

The FC consists of five major combatant subordinate commands, called also pennants:

a. Destroyers Command (DC).

b. Submarines Command (SC).

c. Fast Craft Command (FCC).

d. Amphibious Command (AC).

e. Minesweepers Command (MC).

The number of ships varies from command to command depending on the mission. This organization of the Fleet Command in subordinate commands, names, and number of units are according to Janes Fighting Ships, but the manpower, and some other characteristics are figurative for security reasons. For the same reason, there is not provided any national information, which is considered confidential. Table 3.1 presents the types of units of the fleet command.

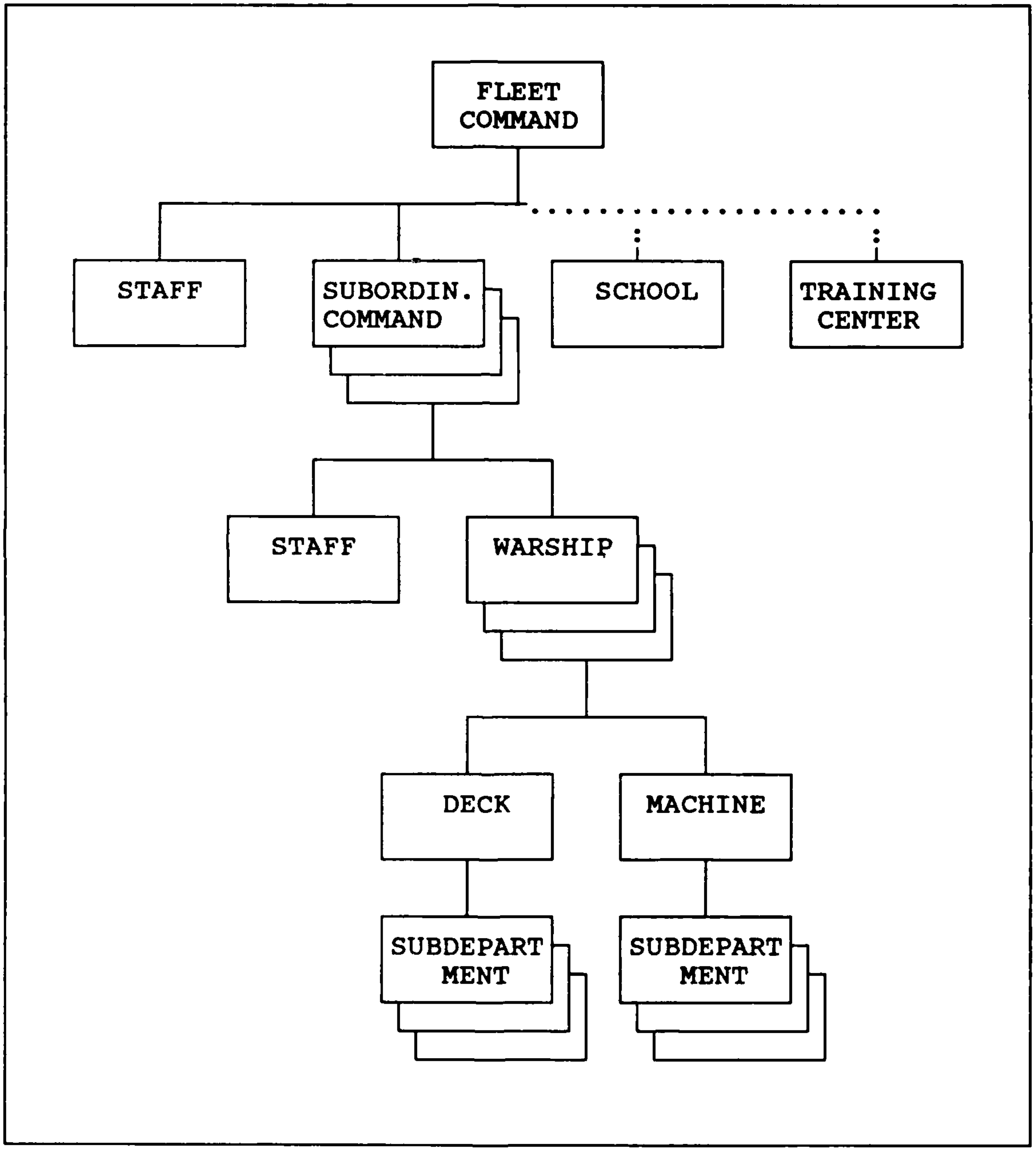


Figure 3.1 Basic Organization of Fleet Command Units

TABLE 3.1 TYPES OF UNITS OF THE FLEET COMMAND

UNIT TYPE	UNIT TYPE DESCRIPTION	COMMAND
FCS	FLEET COMMAND STAFF	FC
DCS DES	DESTROYERS COMMAND STAFF DESTROYERS COMMAND WAR SHIPS	DC DC
SCS SUB	SUBMARINES COMMAND STAFF SUBMARINES COMMAND WAR SHIPS	SC SC
FCCS FAST	FAST CRAFT COMMAND STAFF FAST CRAFT COMMAND WAR SHIPS	FCC FCC
ACS AMP	AMPHIBIOUS COMMAND STAFF AMPHIBIOUS COMMAND WAR SHIPS	AC AC
MCS MIN	MINESWEEPERS COMMAND STAFF MINESWEEPERS COMMAND WAR SHIPS	MC MC

2. Organization of the Fleet Command Officers

According to the Hellenic Navy standards, which do not differ much from the standards used by other countries, the officers in a warship are divided into two major categories. The first category is the Deck Department officers and the second the Machine Department officers.

Table 3.2 shows the basic organization of a warship in departments and subdepartments, and the officer's specialty that corresponds to each of them.

TABLE 3.2 BASIC ORGANIZATION OF WARSHIP

	SHIP=D	
	DECK=D	MACHINE=E
SUBDEPARTMENTS	ADMINISTRATION=D	ELECTRIC INSTALLATION=E
	OPERATION=D	ELECTRONIC EQUIPMENTS=E
	COMMUNICATION=D	DAMAGE CONTROL=E
	NAVIGATION=D	MAIN ENGINES=E
	WEAPONS=D	
	ASW=D	
	SANITARY=M	
	SUPPLY=S	

CODE	SPECIALTY
D	DECK
E	ENGINEER
S	SUPPLY
M	MEDICAL

The Deck Department serves all the needs of a warship from a war machine point of view, and it includes all subdepartments that carry out this task, i.e., Administration, Anti-submarine Warfare (ASW), Combat Information, Communication, Navigation, and Weapons.

The Machine Department serves all the needs of a warship from the movement and repair point of view and it includes all the subdepartments whose functions are related to these purposes, i.e., Electric Installation, Electronic Equipments, Damage Control, and Main Engines.

In large ships only, there are also the Supply Department and the Sanitary Department.

Each warship has a Commanding officer, who is the top supervisor of the ship and belongs to the deck officers. The Executive Officer of the ship is the supervisor of all the subdepartments belonging to the Deck Department, and the Administration Officer. The First Engineer of the ship is the supervisor of all the subdepartments belonging to the Machine Department and the Electric Installation Officer. For each subdepartment there is a supervisor officer, named subdepartment officer. In large ships there is a number of trainee officers. After the assignment process and for the remaining period of the year, the Commanding Officer of a ship is responsible for assigning jobs to subdepartments officers, changing duties between them, if necessary, and informing the HNGS office.

The organization of staff inside a command is similar to the ship organization. Each command consists of the commander, the deputy commander and the staff subdepartments. There are not departments of Deck and Machine but there are

Deck and Engineer Officers. The staff of a command can be treated as a warship during the assignment process.

Among the various officer specialties of the Fleet Command this thesis considers only two of them, the ones that contain the biggest number of officers. These two are the Deck specialty and the Engineer specialty. Most of the ships consist of officers belonging only to these specialties and the other subdepartments are filled by non-officers personnel.

The requirements of rank, specialty, number of officers and distribution of officers, are varied from unit to unit depending on the type, mission, and the size of the unit. As a general rule all combatant ships of the same type have the same organization and requirements.

Officers are assigned to various units according to a position organization table (POT). Table 3.3 illustrates a summarized organization table, which is figurative, for each unit type for the specialty of deck officers. For each specialty there is a corresponding position organization table. Also each unit has its own organization table. Actually the position organization table is more complex since it contains all the positions of units in the Fleet Command with their requirements. Each position in a unit represents a duty and corresponds to one officer who must satisfies the requirements of this position.

TABLE 3.3 ORGANIZATION OF DECK OFFICERS

UNIT TYPE		DECK OFFICERS DISTRIBUTION										TOTAL
		9	8	7	6	5	4	3	2	1	0	
STAFF	FCS	-	-	-	-	-	06	06	01	01	-	16
	DCS	-	-	-	03	03	-	01	01	-	-	08
	SCS	-	-	-	03	02	-	01	01	-	-	07
	FCCS	-	-	-	03	02	01	01	-	-	-	07
	ACS	-	-	-	02	02	01	01	-	-	-	06
	MCS	-	-	-	02	02	01	01	-	-	-	06
	TOTAL	-	-	-	11	11	09	11	03	01	-	46
SHIP TYPE	DES	03	02	03	01	01	-	-	-	-	-	10
	SUB	-	02	02	01	-	-	-	-	-	-	05
	FAST	-	02	02	01	-	-	-	-	-	-	05
	AMP	-	02	02	01	-	-	-	-	-	-	05
	MIN	-	01	01	-	-	-	-	-	-	-	02
	TOTAL	03	09	10	04	01	-	-	-	-	-	27

CODE	RANK	
9	ENSIGN	(ENS)
8	FIRST LIEUTENANT	(1LT)
7	LIEUTENANT	(LT)
6	LT COMMANDER	(LCDR)
5	COMMANDER	(CDR)
4	CAPTAIN	(CAPT)
3	COMMODORE	(COMD)
2	REAR ADMIRAL	(RADM)
1	VICE ADMIRAL	(VADM)
0	ADMIRAL	(ADM)

D. ASSIGNMENTS MECHANISM AND CRITERIA

Up to this point the organization of the officers and the organization of the units in the Fleet Command have been discussed. Now the officer assignment mechanism as well as the criteria that affect this mechanism will be described.

1. Assignments Mechanism

All officers up to a certain rank are assigned to various units. The goal is to select the right officer for the right job in the right unit in the best objective manner.

All officers through the rank of Lieutenant should be assigned to warships which are part of each subordinate command. The staff schedules the assignments of the officers up to the rank of Commander, which is the highest possible level rank that can exist in the warships. The assignments of the other ranks and their criteria follow a different process which will not be discussed as subject in this research. However this research includes the most appropriate information that can be viewed and used in the assignment process of these excluded ranks, but does not include the process description.

The officer assignments process is closely related to the officer promotions process. The promotions usually occur in the period of May and June of each year.

After the official promotions the staff schedules and processes the annual assignments separately for each corresponding rank, by specialty. The staff office determines the officers who meet the requirements for assignment, and the new unit, providing also any requested information about the officers.

2. Assignments Criteria

There are certain criteria that affect the mechanism of scheduling the assignments. Among them the most common criteria have been chosen, so that they will be easily applied to every command of the HNGS with minor modification.

a. Job

The first criterion is the job. It determines the purpose of an officer in a warship. Even though a job is common to all warships, it does not means that the same officer can be assigned arbitrarily to any one of these ships.

b. Specialty

There are several specialties in the Hellenic Navy but four of them can be found in the Fleet Command: Deck, Engineer, Supply, and Sanitary. As stated before the Deck officers and the Engineer officers compose the main officer manpower of each unit and, of course, of the Fleet Command. Table 3.2 shows the specialties that correspond to each job in a warship. Table 3.3 determines the number of Deck officers assigned to each unit type. For each specialty there exists a similar table.

c. Rank

The third main criterion is the rank. Table 3.3 determines the number of officers per rank assigned to each unit type for the Deck specialty. Each job position of each unit corresponds to a specific rank code, but sometimes this is not compulsory. This is an exception that happens in the case that the number of officers of the specific rank after the promotion process, is less than the number of positions that requires this rank code in the position organization table.

All students of the Naval Academy right after their graduation take the initial officer rank named Ensign, and they are assigned to destroyers for one year of training. After this training they are assigned to the general education school (GES) for general education. They remain there for one year and then are assigned to special education school (SES) which provides a special education for their later duties in the warships. They remain in SES for one year.

During the promotion process they are promoted to the rank of the 1st Lieutenant. During the assignment process they are assigned to the various ships, in which they serve in the positions of 1st Lieutenant according to the organization table, with the duty of supervisor in a subdepartment.

Lt. Commanders and Commanders officers can be assigned to staff positions.

d. Service Time

There is a minimum time that an officer must serve continuously without new assignment given as follows:

1. Ensign, 1 years
2. First Lieutenant, 1-2 years
3. Lieutenant, 1-3 years
4. Lt Commander, 1-2 years
5. Commander, 1-2 years

The maximum service time depends on the promotion to the new rank, and is not determined directly. The rule is to avoid assignments of officers who do not satisfy this minimum period.

e. Education

Military education is combined with the schools in the rank criterion. Officers with special civilian education can be selected for special purpose positions, or missions, especially outside the fleet command. In this analysis only the education that corresponds to the Ensign rank is provided.

f. Order in Class

This criterion is examined whenever two or more officers have the same qualifications and belong to the same class. In this case officers with better order are given preference.

g. Year of Class

This criterion determines the subset of all officers who belong to the same class. Since a rank contains more than one of these subsets, the year of class determines also the order of each class inside the same rank. This criterion in combination with the criterion of the order in class are important tools, because the assignment process makes use of the information in the order of an officer inside the same rank as well as inside the same class.

h. History

There exist records for each officer, containing all personal and service data. Information about assignments, promotions, and education, of an officer are maintained in historic files. This data must always be kept up-to-date, because they reflect the real picture of an officer and provide scheduling personnel with the required information to accomplish their task.

E. PROBLEM SOLUTION

From the analysis up to this point, one may conclude that the manual system of the assignments process inside the Fleet Command is very complex, inefficient, and time consuming, as well as the whole system concerning officers data. The staff tries to discharge its responsibility by working very hard continuously, working with construction, examination, classification, reconstruction, and reexamination of officers records, scheduling the assignments and providing any requested information.

A solution that can overcome these problems is the development of a database system by computer support through the use of a microcomputer. There are two database processing systems: the file processing system and the database processing system. Between these two, the second one is selected for the reasons that are explained in the previous chapter.

Also the use of a microcomputer is a cost effective solution. The cost of buying, installing, maintaining and changing the system (hardware, DBMS) is very low. Since the system is automated, a reduction of the staff's manpower is likely. This reduction is very important feature of the new system because it saves personnel, making it available in other vital positions. As an effective beginning of building the new officers database system, the Stand-Alone microcomputer database presents few problems. Furthermore, the application can be solved easily with the relational database model.

In conclusion, a database processing system developed on a microcomputer will be shown as an efficient and cost effective solution to the officer assignment problem.

F. SYSTEM GOALS AND REQUIREMENTS

1. System Goals

Previously the reasons of why the author is interesting in switching from the existing manual system to an automated have been stated. These reasons lead to the goals, that is, targets for achievement. The following targets must be achieved by the system:

a. To achieve greater processing speed. The computer's inherent ability to calculate sort, and retrieve data and information is faster than people doing the same tasks, as well as, easier. This is a very important factor for a decision-oriented processing environment.

b. To achieve accuracy in data retrieval. This can be achieved if the criteria for job assignments is carefully described and taken into account in the application programs. In the case of reports and lists the possibility of information omission is eliminated.

c. To achieve better quality of decisions. Up-to-date data/information can be made available to decision makers.

d. To achieve faster information retrieval. Locating and retrieving information from the storage repository is faster.

e. To improve productivity by reducing manpower. This is very important since we can reduce the staff personnel involved in the manual system and use them for other productive tasks.

f. To improve the security level regarding personnel information against unauthorized users called intruders, who want to read these information or change data.

2. Requirements

Since we have defined the goals of the systems, we also should specify the requirements that the system must satisfy. That is, the capabilities that the system must provide as project tasks so that the previously defined goals can be achieved. The new system:

a. Must be able to store any information about officers of the Hellenic Navy that is currently stored on papers.

b. Must enforce reliability, i.e., it must be able to perform its intended function under stated conditions for a stated period of time.

c. Must be able to provide any stored information upon request.

d. Must include sufficiently defined criteria so as to provide solutions for the assignments as accurately, effectively, and objectively as possible.

e. Must be easy to use, by personnel without special programming and computer knowledge or skills.

f. Must be cost-effective.

g. Must be useful.

h. Must provide feature extension.

G. INPUT AND OUTPUT INFORMATION

Before describing the design phase it is desirable to describe the required inputs and outputs. This can help to better understand and organize data into the supporting system files. It is not specified in details what the exact input and output information will be but some insights and understanding are provided. These thoughts should be taken as hints and guidelines concerning system input and output information for the product design, but not as rigid requirements.

1. Input Information

Since the system is intended to deal with officers, the required input information should be considered officers' data. Some basic data that can be viewed are the following:

a. Each officer has a unique serial number, rank, specialty, nomination date, promotion date in the current rank, an order in its class, and a home city.

b. Each officer serves in some unit since a certain date (enrollment date), and has been assigned a job (duty). The unit is identified by a unique name, and belongs to a command.

c. Each officer has some education, military and non-military.

d. Each officer can speak or not a number of foreign languages.

e. Each officer has some historical data about previous assignments, duties, promotions, education, etc..

f. Each job position has some requirements, which the officer should satisfy in order to be assigned in this position.

2. Output Information

The required output information that the system should provide are the following:

a. Scheduling officers' assignments by rank.

b. List of officers assignments of a requested rank providing serial number, name, rank, source unit, destination unit.

c. List of the officers who serve in a specific unit, their rank, and enrollment date.

d. List of all officers in a given requested order.

e. List of Commanders, Commanding officers, Executive officer, and First engineers with their service unit.

f. List of the officers of some requested rank with a status form (rank, name, specialty, unit, duties, and enrollment date).

g. Service time report (assignment and promotion history) for each officer including rank, units, order, and enrollment dates.

h. Present status report of each officer.

H. SUMMARY

System analysis is the process of gathering and interpreting facts, diagnosing problems, and using the facts to improve the system through better procedures and methods.

There are six naval personnel administration functions: Procurement, education and training, assignment, treatment, promotion, and separation. The research is focused in the assignment process.

One of the major responsibilities, which also is the biggest problem for the decision makers, is to schedule the annual assignments of the officers to warships inside the Fleet Command. This is the job of the assignment scheduling committee (ASC).

Two major organization components compose the assignment process: The organization of the Fleet Command units, and the organization of the Fleet Command officers. The products of these two components are the position organization table with their requirements, as well as the officers who must be assigned to each position.

The officers assignments process is closely related to the promotion process. The promotions usually occur once a year for each rank. The promotion process after the annual promotions acts as a force to the balanced system and the result is a change to an unbalanced system. The goal is to select the right officer to the right position in a unit, satisfying the requirements and preserving the balance of the system.

The job of the assignment scheduling committee is to react with an opposite force in order to get the balanced system again. That is, the ASC through the assignment processing must remove officers who do not satisfy the unit position's requirements and to reassign them in other proper positions.

There are certain criteria that affect the mechanism of scheduling the assignments which are considered components of the reaction force. An intelligent decision support database system developed on a microcomputer will be shown as an efficient and cost effective solution to the officer assignment problem.

There are certain goals, that is targets, that the system must achieve, as well as certain requirements, that is capabilities, that the system must provide as project tasks so that these goals can be achieved.

Finally, this chapter presents the system input and output information which will compose hints and guidelines for the next phase of the development process, that is the system design.

IV. SYSTEM DESIGN

The design of a system is the proposed solution to the problem, that is, the translation of requirements into ways of meeting them. The design process of the system includes two phases: the logical design phase and the physical design phase [Seen, 1984:p. 224]. First, the logical design is developed. Once the logical structure has been defined, it is transformed into physical form for implementation using a specific DBMS.

In this research, the design produces the details that state how the system will meet the requirements identified during system analysis in Chapter III.

In particular, the discussion of objects and their relationships to database structure have been evolved from work based on the entity-relationship (E-R) data model. The E-R model was chosen as a representive of the class of the object-based logical model. This model was chosen since it has gained acceptance as an appropriate data model for data base design and it is widely used in practice [Korth & Silberschatz, 1986:p. 6].

A. LOGICAL DESIGN

Logical design is the process that describes the system functions, relation diagrams, relation definitions, relation scheme, and data dictionary [Kroenke & Dolan, 1988:p. 167]. This is accomplished by examining the entities and identifying the relationships among them, building the entity-relationship (E-R) diagram and transforming it into normal relations according to the relational normalization theory.

1. Processing Control Mechanism

One of the system requirements, as stated in the previous chapter, is that the system must be easy to use by personnel without special programming and computer knowledge or skills. Even more, the system performs a logical sequence of functions and subfunctions and each logical sequence generates a path of actions. Therefore, one processing control mechanism must exist which will direct and control the database processing system.

A menu-driven interface is such a processing control mechanism. It consists of a sequence of menus and submenus which direct the process to the appropriate operation or action to be executed. Figure 4.1 illustrates the structure of a two-level menu driven system.

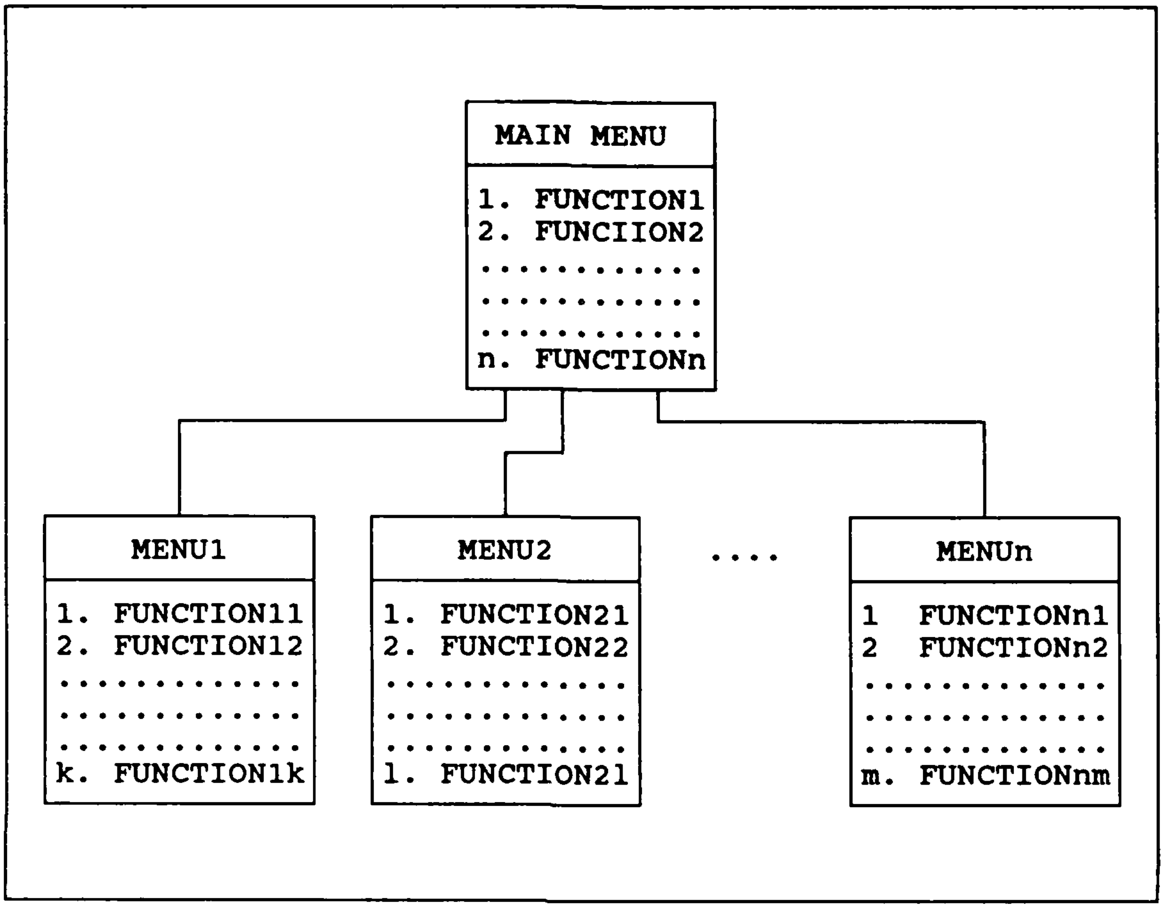


Figure 4.1 Architecture of Menu-Driven System

2. Classes of System Functions

The system will perform a number of functions which are divided into four main classes.

a. Update Data

The update data function performs inserting, deleting, and editing (modifying) data in the database. The user of the system should be able to insert a new record, to delete an existing record, and to modify records, in all the permitted supporting databases. However, there exist a few files, which are automatically updated, based on changes to the databases. This function takes place anytime. Also there are some databases that are not accessible by the user, as for example, the position organization table (POT), the user's authentication and the user's access tracking.

Each update operation has as a result a sequence of other update operations which have to be executed immediately, and automatically. So, a deletion operation to a record in the database, may consist of a sequence of insertion, deletion, and modification operations in other database tables. This means that it is a very difficult and complicated process for the system to make use of any assistant menu that a DBMS may provide for this function. For this reason the system generates its own update data function, which is considered an application process using the commands of the DBMS. Otherwise it is possible for the system to provide wrong results.

b. Assignment Process

This component performs the officer assignment function. Because of the complicated job and the various criteria of the assignments, the system should be able to schedule the assignments as they actually take place, that is, separately per rank for each specialty. The type of the assignments is selected from an appropriate submenu.

c. Lists and Reports Production

A number of application programs support this function which retrieves the necessary information about officers from the database repository and produces the appropriate lists and reports upon request.

d. System Access Security

System access security is very important since the system works inside a military environment and the security of information is a critical part of the military. The system in practice contains classified, confidential and secret information for both the officers and the units. So the system access must be strictly controlled.

(1) User Authentication. This is a military way for authorization validation. System access security should be provided at the system start-up, through a password control mechanism. Whenever a user attempts to access the system, the system checks the authentication by asking him to enter his password. The authorized relation between users and valid passwords are contained in a file. Only valid passwords can access the database system. This way provides a security level against the unauthorized users or intruders who try to access the officers' database.

(2) User Access Tracking. Although the above function of user access security satisfies the requirements of the military security rules, an additional function is applied to the system called "tracking." The user, the modifications that have been done to the supporting system files, and the time this occurred are recorded. Every time a valid user performs a task, a record is automatically created containing the time, the date and the kind of task. This provides an audit trail of who, what, and when of an operation, allowing us to find out easily what exactly happened, for any examination process about the system's violation.

e. Historical Data

This function performs historical operations which illustrate the officer's past action. For each officer's career transaction as assignment, promotion, nomination, education, or deletion, a record is automatically created to the corresponding historical file. This provides a tracking of each officer's career, and provides a very useful, document to decision makers for future decisions.

3. Entity-Relationship (E-R) Process

To develop the proposed system and before going into relational database process, it is needed to identify the entity-relationship diagram of the design. The entity-relationship (E-R) diagram is a diagram which shows individual entity occurrences and their relationships. Modell [Model, IEEE, 1985:p. 123] referred to the E-R diagram model as one of the most effective methods of analysis. He concluded that the analyst was able to construct a meaningful model of the real world based upon his interpretation of it.

The convention which will be used in drawing an entity-relationship diagram is that entity types will be represented by rectangles and relationships by diamond-shaped boxes. Connecting lines show which entities are associated by each relationship type.

Following Howe's process [Howe, 1983], there are two different ways in which an entity type can participate in a relationship. First some of the enterprise rules insist that every occurrence of an entity participates in the relationship. This situation of entity's membership class is termed "obligatory" or "mandatory." Second, other enterprise rules allow occurrences of an entity to exist independently, without inclusion in the relationship. This situation of entity's membership class is termed "non-obligatory" or "optional." A dot inside a stripe on an entity symbol means that the entity's membership class is obligatory. A dot outside an entity symbol means that the entity's membership class is non-obligatory. For a relationship between two entity sets there are then four possible combinations of membership classes. These membership classes of entities are very important because they influence the process of translating the E-R diagram into the relational model and defining the schemes.

Appendix A illustrates the E-R diagram which corresponds to the entities and their relationships. For better illustration the diagram, the same figure includes as many entities with their relationship as possible. This E-R diagram provides a number of relationships, which correspond to the diamonds.

4. Database Relation Scheme

The entity sets in which the office of HNGS officers management system is interested and the relationships among these entity sets have been established as in Appendix A, through the entity-relationship diagrams. These diagrams compose the skeleton which will be translated into relations, generating a normalized relation scheme for the database system.

The relation schemes can be built according to relational theory and the rules of functional dependency and normal forms. Chapter III gives a brief description of this important theory and these two fundamental rules.

Having as guidelines the relational theory and the rules of functional dependency and normal forms, and following the methods that the references describe about translating the entity-relationship diagram into relations, the relation schemes of the Fleet Officers database were generated as shown in Appendices B and C.

Appendix B illustrates the process of transforming the entity-relation diagram into relations, and Appendix C illustrates the functional dependency of these relations. Also, the data dictionary in Appendix D illustrates the relational database dictionary structure of the proposed system design.

5. Relation Definition and classification

The relation scheme consists of two kinds of relations: the entity relations and the relationship relations. Both of them become the relational database of the Hellenic Fleet Naval officers. These relations can be classified in categories.

a. Officer Category

This category consists of the officer's relation which contains the required information for all officers in the Fleet Command, who are on active duty. Each officer serves in one unit which can be ship, staff, school or out of the Fleet unit.

b. Unit Category

This category consists of those relations which compose the units in which an officer can serve.

(1) Command. Command contains information for all the existing Commands that compose the active war-field of the Hellenic Fleet. One of the warships in a command is the base of the Commander, and it is called the commanding-ship.

(2) Fleunit (Fleet Unit). Fleunit describes each unit of the Fleet. The Fleet consists of two unit types: the warship or simply called ship, and the staff. All ships of the same type belong to the same Command.

(3) **School.** School gives information about the military and civilian schools in which officers can be assigned for one year during their career in the Fleet.

(4) **Othunit (Other Unit).** Othunin contains units that do not belong to the Fleet.

(5) **Organic.** Organic represents all organic positions of Fleet, that is, all the active positions with their appropriate requirements for each fleet unit. Officers can be assigned to these positions in order to fill the organic positions table of each fleet unit.

c. Assignment Category

The assignment category consists of those relations which contain information about the current assignments of officers, as well as the reassignment of an officer because of the promotion process

(1) **Power.** A relationship between OFFICER and FLEUNIT contains the officers that have been already assigned to the Fleet units (that is staffs and ships). This relationship represents any time after finishing the assignment process the actual situation of staffs and ships and officers. More specifically, it contains information such as which officer serves in which Fleet unit? what is his duty? when assigned to this unit?, etc..

(2) **Study.** Study is a relationship between OFFICER and SCHOOL. It contains all officers who have been assigned to a school unit as students.

(3) Oof (Out of Fleet). Oof is a relation between OFFICER and OTHUNIT which contains the Officers who have been assigned to units that do not belong to the Navy Fleet organization.

(4) Assment (Assignment). Assment is a relation between the OFFICER and units FLEUNIT-SCHOOL-OTHUNIT. This is actually a relation which contains what the assignment process tries to accomplish. It contains information about the officers to be assigned to units after the promotion process, that is the annual assignments. More specifically, it contains the subset of officers who are going to be removed from one unit to another new unit, according to the assignments criteria. It constitutes the order of the Officers assignments that the staff issues to the units (staff, ship, school, non-fleet unit). It is important to remember that the assignment process is performed separately for each rank and specialty. At this point the assignment processing finishes. From this time on, officers change positions and both, the old and the new units, inform the staff.

d. Education Category

Education category consists of those relations which give all the information about the education of the officers.

(1) **Edution (Education).** Education contains information about the military and civilian education of each officer.

(2) **Forlang (Foreign Language).** Forlang contains the officers who speak foreign language with the knowledge level. An officer may be capable of speaking many languages.

e. History Category

History category consists of those relations which provide a tracking of each officer's career.

(1) **Asshist (Assignment History).** Asshist records all assignments of each officer which have taken place during his career.

(2) **Prohist (Promotion History).** Prohist keeps track of all officer's promotions which have taken place during his career.

(3) **Nomhist (Nomination History).** Nomhist contains information about the nomination of each officer.

f. Security Category

Security category consists of relations which support the security of the system's access.

(1) **UserPas (User Password).** Userpass contains all the users names who are authorized to use the system and the corresponding valid password to each user.

(2) **Userlog (User Log On).** Userlog keeps data about the user and his activity on the system with the corresponding date and time.

6. Relation Database Dictionary

The system's database dictionary contains additional information about the database structure, that is it contains data about data. The database dictionary of this research consists of three major parts: the relations database structure, the domain definition, and the domain values.

a. Relation Attribute Structure

The relation attributes structure contains the relation name, the key, and items of four columns information. The first column is the item serial number. The second contains the attribute or field name. The third specifies the corresponding domain in which the attribute can take values. The last column, that is the fourth column, describes the meaning of the field.

b. Domain Definition

The domain definition part describes the domain name, the type and the width. The type of the domain can be character (C), date (D) in format MM/DD/YY, numeric (N), logical (L) T or F, and memo (M) for large blocks of text.

c. Domain Value

The domain values part contains specific values of the corresponding domain definition which are not provided in the definition part.

Appendix E describes the database dictionary of the proposed system design.

B. PHYSICAL DESIGN

Physical design is the second stage of the database design, and is a stage of transformation that translates the logical scheme into the particular database constructs for the DBMS to use.

The data dictionary of the system in Appendix D contains the appropriate information which defines the guidelines in developing the physical database design, from the logical design of this chapter.

1. DBMS Specifications

Today, database management systems built for microcomputers have become very popular. They provide an inexpensive and easy way for developing database systems. In order to build the officers database system on a microcomputer environment, the user on the micro must be able to access, somehow, the database on a larger computer. The availability of a programming language to write application programs is also a necessity.

dBASE III PLUS is as a popular relational database management system. This software helps to create and maintain a relational database system for microcomputers, under one of the popular operating systems. It includes both a data manipulation language and a general purpose programing language which offers a number of ways to manage information. For these features, the thesis uses dBASE III PLUS as an appropriate example of the DBMSs that can support the system design.

The most important features, as well as, the limitations of dBASE III PLUS, according to Simson [Simson, 1986] and Jones [Jones, 1987], are provided as follows.

a. **Features of dBASE III PLUS**

1. Program and data independence. Changes in file structure do not affect application programs.

2. Updatability. Data can be easily updated.

3. Date and memo data types. Besides the known common data types, that is, character, numeric, and logical, dBASE 3 PLUS provides two more powerful tools: "Date" date type for managing dates, and "memo" data types for managing texts.

4. Information saving. It saves information as disk files in nine specialized formats each serving a specific dBASE 3 PLUS processing need.

5. Built-in high level language. It is extremely powerful and supports structured programming.

6. Interface capabilities. It allows interfacing with other software systems, such as Super Calc, Symphony, Wordstar.

7. Multi-User local area network (LAN) capabilities. This enables each dBASE 3 PLUS user to establish and to attach to a local area network.

8. Simultaneous file access with data protection. This provides file lock and unlock, as well as, record lock and unlock.

9. Control access data. It provides password protection at eight levels which can be defined for groups, users, files, and fields.

10. Sorting and indexing capabilities.

11. Stand-alone operating capabilities.

b. **Limitations of dBASE III PLUS**

1. Number of records in each file. Each database file can have up to one billion records maximum. The maximum size of each file is two billion bytes.

2. Number of fields in each record. Each record can have up to 128 fields. The width of each field can be up to 4,000 characters maximum.

3. Number of database files open at the same time. It allows up to ten database files to be opened at the same time, or fifteen files of all types . Seven index files and one format file can be opened for each active database file.

4. Length of file name and field name. Filenames can be up to 8 characters long, and fieldnames can be up to 10 characters long.

5. Active memory variables. The maximum number of active memory variables is 256. The total number of bytes for memory variables is 6,000.

2. **Hardware Specifications**

Since the thesis uses dBASE III PLUS as the appropriate example of data base management system (DBMS) to manage the information, there are also some hardware specifications which support the software specifications.

a. Microcomputers. 16-bit microcomputer IBM PC, PC/XT, PC/AT, 3270 PC or 100% compatible.

b. Minimum memory. 256 KB RAM for stand-alone operating mode but 320 KB or more is recommended for best use. In local area network operating mode, memory requirements for machines is 640 KB.

c. Disk capabilities. One disk-drive is possible, two disk-drives with one hard-disk are suggested.

d. Printer. Any printer of 80 columns, which can support the above microcomputers is suitable.

e. MS-DOS or PC-DOS version 2.0 or newer.

f. Any monochrome or color monitor.

C. SUMMARY

System design is the process of planning a new system to replace or complement the old. It is a solution that translates the requirements into ways of meeting them. The design process of the system includes two phases: The logical and the physical design phase.

The system can perform a number of functions which are divided into five main classes: Update data, assignment process, lists and reports production, system access security and historical data. Menu-driven system is the processing control mechanism which directs and controls the database processing system.

For system design the entity-relation (E-R) model as a representive of the class of the object-based logical model has been chosen, since it has gained acceptance as an appropriate data model for data base design and it is widely used in practice. Appendix A illustrates the E-R diagram which corresponds to the system's entities and their relationships.

The entity sets that are of interest to the application, as well as the entity-relationship diagram among these entities form the basis from which the relations and the normalized relation schemes for the database system are derived. Appendix B illustrates the process of transforming the E-D diagram into relations, and Appendix C shows the functional dependency of these relations.

The system's generated relations are classified into six major categories: Officer, Unit, Assignment, Education, History and Security. The system's database dictionary contains additional information about the database structure, that is data about data. Appendix E describes the relations database dictionary.

The physical design is the stage of transformation in which the logical design is transformed into the particular database constructs of the DBMS. dBASE II PLUS is used as an appropriate example among the various DBMSs, which can support the last phase of the system's development process, that is the implementation phase.

V. MODEL DEVELOPMENT AND SYSTEM IMPLEMENTATION

The structured system analysis and design for the proposed system have been discussed in the previous sections. Annual assignment processing is the most difficult and the most important part of the system. This chapter presents the assignment model development and the system implementation.

A. ASSIGNMENT MODEL DESIGN

During the assignment process the necessary criteria are inserted to the model system, they are examined, evaluated and processed by the system. The output gives the annual assignments for a specific rank which is the first parameter of the system.

The assignment model design establishes a real model which can exist by itself. It is based on real situations, in the military management functions, thus making the process very complex. The assignment model has its own criteria, rules, and strategy which establish its existence in a real military environment.

1. Model States

The states of the model determine its balanced or unbalanced situations before the promotion process, after the promotion process and before the assignment process, and after the assignment process.

a. **Initial State "A"**

Initial state "A" represents the situation of the system before the promotion process and is in a balanced state. During this stage every position in the position organization table (POT) is atomic. That is, each officer satisfies the requirements of the position of the unit in which he serves.

b. **Middle State "B"**

Middle state "B" represents the situation of the system immediately after performing officer promotions for each rank and is a temporary state. The promotion process acts as a force to the system in initial state "A" and the system now changes to middle state "B", which becomes an unbalanced state at this time. During this state there are officers in the system who do not satisfy the requirements of the position organization table. The corresponding positions called non-atomic and the system must be transformed again into a balanced situation.

In addition, the promotion process affects the annual compulsory schools of each rank. This affects the assignment process because the officers from the schools must be assigned to the organization positions inside the Fleet units, as well as officers from organization positions must be assigned to schools.

c. Final State "C"

Final state "C" represents the situation of the system after the assignment process, and is a balanced state. Because of the promotion process, the assignment process reacts as an opposite force to the system in middle state "B" and the system changes to final state "C". This state is equivalent to the initial state "A" and all the positions in the position organization table are transformed again to atomics. That is, each officer satisfies the requirements of the position in the unit in which serves.

2. Balance Transformation Process

As it is obvious, the significant point is the transformation process of the system from middle state "B" to final state "C". This is the task of the assignment processing. During this process, officers who do not satisfy the organization position requirements of the unit in which they serve must be moved into the appropriate unit positions.

All the possible positions of the units in which officers serve and may not satisfy the requirements represent the assigned positions (ASSPOS) of the model system. All the possible officers who can be assigned to these positions represent the assigning officers (ASSOFF) of the system. The system's ASSOFF must be moved and matched with the system's ASSPOS, generating pairs, so that in each pair the ASSOFF satisfies the requirements of the ASSPOS.

The problem results because of the complexity of the system. It has components of military criteria and rules of assignment which restrict and make difficult the problem's solution under these situations. The system should follow the military hierarchy strategy, and it must avoid aimless assignments. Another component of the complexity is the large number of officers.

After the above description, certain questions are generated regarding the model's development. Can the system be transformed from an unbalanced stage into a balanced stage? As an example, how can the system choose from 10,000 officers who can serve in 500 units, those officers who don't satisfy the position requirements of the unit in which they serve? If the system finds 4,000 of these officers, how the system can assign these officers to appropriate positions inside the units, so that they must satisfy the position requirements? How can the system use the assignment criteria and how can the system obey the military structure rules? These are some questions which the established model development will be able to solve under real application conditions.

3. Rule of Assignment

The rule of assignment, as well as the assignment criteria, composes the autonomic existence of the model. The assignment model development follows a sophisticated methodology, based on the assignment criteria, as well as to some esoteric structured rules of the position organization table (POT) which compose the organization of the Naval Fleet. In addition, it consists of a logical sequence of actions that must be followed, and it must obey the basic rules of minimum movements if required and if possible, the military hierarchy rule or normal rule, and the rule of equivalent levels.

a. Rule of Military Hierarchy

This is the most important rule that the process follows since this rule determines assignments according to the military hierarchy process. That is, officers must occupy positions that are occupied by officers of equal or greater rank. This means that the lower levels try to push the upper levels. Three criteria are used to build the hierarchy officers' distribution: Rank, class year (clayear), and class order (claorder).

The set of all the possible subsets that the hierarchy rule can be applied, is defined as : [(CLAORDER), (CLAYEAR), (RANK), (CLAYEAR, CLAORDER), (RANK, CLAORDER), (RANK, CLAYEAR), (RANK, CLAYEAR, CLAORDER)]. The assignment process in order to distinguish officers inside a level uses any element of this set.

b. Rule of Minimum Movements

Since each movement incurs expenses, the process must be able to minimize these movements without upsetting the system's balance. This means that an officer O1 can be assigned to a position P1 inside the same unit U1, if and only if, this officer O1 satisfies the position's P1 requirements and this position P1 is occupied by another officer O2 who serves in the same unit U1 and does not satisfy the position's P1 requirements. That is, the process follows the path of minimum movements, and the rule of minimizing expenses.

c. Rule of Equivalent Levels

Two consecutive levels of assignment hierarchy L1 and L2 are considered equivalent if there is not any reason for the officers of the lower level L1 to be assigned in positions of the higher level L2. The positions of the two levels are also considered equivalent. As a result the lower level L2 is a steady level and the officers in this level stay in the same positions.

B. ASSIGNMENT MODEL STRATEGY

This section describes the set of all the possible functions that exist in the assignment processing and affect its strategy. After defining this set, the implementation process is simpler. That is, for any given rank there exists a subset of these functions which forms the implementation part for this specific rank. Simply stated, the process implements only those functions that are related to that given rank. The whole process consists of four major phases:

1. Selecting data.
2. Processing data.
3. Checking data.
4. Providing result.

1. Selecting Data Phase

The selecting data phase is very important, since this phase creates the sources of data. During this phase the process selects and isolates for a given rank all the data concerning organization position requirements of the units and officers under assignment who meet all the requirements and are being scrutinized in the assignment process. This phase consists of two basic stages creating two major data sources for each specialty. The final result is that, by moving all required data into fewer and smaller files, the overhead of the search operation is reduced, and the processing is speeded up.

a. Assigned Position Stage

This stage creates the ASSPOS source which contains the set P of all possible organization positions of units for a given rank in which officers do not satisfy the position requirements, including their requirements and the officers who occupy each position. This set consists of two subsets: The subset P1 of the positions that are created because of the promotion process, and the subset P2 of the positions that may be created during the assignment process. That is, the process excludes all unit organization positions in which the officers who serve do not satisfy the criterion of minimum service time.

There is no case in which there are vacant unit positions because it can happen to have the number of active officers exceed the number of the total unit position requirements of the position organization table. However the system also checks for vacant unit positions and if it finds such, it inserts them into ASSPOS source data.

b. Assigning Officers Stage

This stage creates the ASSOFF source which contains the set of all possible officers O for the given rank who can be assigned in the above positions, including their data and the unit position for each officer. This set consists of two subsets: The subset O1 of officers who don't satisfy the position requirements because of the promotion process, and the subset O2 of officers who may change their requirements during the process. The ASSOFF source, set O, contains also those officers who come from the STUDY and the OOF relations.

2. Processing Data Phase

This phase receives data from the source phase, that is, the selecting data phase. During this phase the selected data are examined, evaluated and processed according to the rest of the assignments criteria and to the processing rules. The process moves each officer from ASSOFF and matches him with exactly one position in ASSPOS, creating an officer's assignment to a position in a unit in the best objective manner. The whole processing data phase consists of a logical sequence of stages.

a. Hierarchy Stage

This stage builds the hierarchy of officers and positions. It uses the ASSPOS and ASSOFF sources. The process sorts or indexes the two sources in the sequence fields of RANK, CLAYEAR, and CLAORDER. The result is two hierarchical data sources which use the remaining process. The hierarchy stage gives the hierarchical distribution of officers in the ASSOFF source, and of positions in the ASSPOS source. This task of distribution is divided in three levels which are created with conditions inside the engagement rules.

The higher level (HL) contains the unit positions which are occupied by officers who were just promoted from this given rank to the next rank. Thus officers of the next rank occupy unit positions of their previous rank level. These positions belong to the ASSPOS relation and must be matched with officers of this rank.

The middle level (ML) contains the officers of this rank who were not promoted but stay in the same rank. This level consists of two sets of officers. One set of officers is common to both relations. That is, the officers belong to the ASSOFF and their positions belong to ASSPOS. The second set contains the officers who belong only to

ASSOFF and their positions do not belong to ASSPOS. From the hierarchy aspect this level is divided into two hierarchy sublevels, which are created dynamically during the process. The middle upper level (MLU) which contains the major officers of the middle level (ML) who will be assigned to positions of the higher level (HL) according to the hierarchy rule. The middle lower level (MLL) which contains the remaining officers of the middle level (ML). A subset of these officers will not be assigned in any position, because of applying the equivalent levels rule. The remaining officers will be assigned according to rules of hierarchy and minimum movements.

The lower level (LL) contains the officers who were just promoted from the previous rank to this given rank. These officers belong to the ASSOFF source of this rank but occupy unit positions of their previous rank. They must be assigned to upper positions which belong to the MB level.

b. Trainees Assignments Stage

This stage performs only assignments of the new Ensigns who have graduated from the Naval Academy according to the normal hierarchy rule. In any other case this stage is omitted. All new officers are assigned to Destroyers for one year with the duty of trainee.

c. Students Assignments Stage

This stage performs the assignments of officers of any rank to the annual schools because of the promotion process. If a rank does not provide any school then this stage is omitted and not executed.

d. Staff Assignments Stage

The stage performs the assignments of officers to staffs. Each time only officers who belong to a specific class year or come from a specific annual school can be assigned to positions inside the staffs. If there are not staff positions for the given rank this stage is also omitted. The stage uses the hierarchy rule.

e. Specific Assignments Stage

This stage performs assignments of officers to a specific job, for example, Commanding Officers in Destroyers. Each time only officers who belong to a specific class year or come from a specific annual school can be assigned to a specific job. If there is not such a case for the given rank this stage is also omitted. The stage uses the rule of minimum movements and the rule of hierarchy.

f. Medium-to-Higher Stage

This stage is applied to all ranks and performs the assignments of ASSOFF source officers middle lower level to the ASSPOS source positions higher level (MLU -> HP). The stage uses the normal hierarchy rule. The process selects the position from the ASSPOS which is occupied by an officer of next higher rank. It then finds an officer from the ASSOFF who has different job and assigns him to this position. The stage takes place for the rest of the officers after performing the above assignments stages.

g. Medium-to-Medium Stage

This stage is also applied to all ranks and is executed after the higher-to-medium stage. It performs the assignments of the rest of the officers in the middle level.

First it uses the rule of equivalent levels. It selects the officers of the steady sublevel from the ASSOFF middle lower level (MLL) as well as their positions from the ASSPOS middle lower level (MLL) and deletes both of them in the appropriate place. Next it applies the rule on the normal hierarchy to the rest of the officers of the middle level and assigns them to Fleet units.

h. Lower-to-Medium Stage

This stage performs the assignments of ASSOFF officers of the lower level (LL) to ASSPOS positions in the middle level (ML). This stage is divided in two substages. First it applies the minimum movements rule. It tries to assign an officer of the lower level (LL) in position inside the same unit if there is an available position in the middle level (ML). Second it applies the hierarchy rule. The process selects the rest of the positions from the ASSPOS source middle level (ML) which are occupied by officers of this rank who were assigned during the previous stages. It then finds an officer from the ASSOFF source of lower level (LL) and assigns him to one of these positions.

3. Checking Phase

This phase checks the results of the assignments and is common for all ranks. It actually checks to see if there are still available officers who have not been assigned. The process uses the ASSOFF source data and tries to find if there is any officer inside. If found, then the process asks the user and assigns the officers to the Fleet Command or to the Headquarter General Staff according to the user's answer.

There is no case in which there are still available positions for a rank since every year the number of candidate officers exceed the number of the manpower requirements. However the system checks the ASSPOS source data if there is any available position. If the system finds such, it informs the user. The officers who are assigned during the checking phase are controlled by the corresponding staff. At this point the assignment process terminates and the system goes to the result phase.

4. Providing Result Phase.

This phase produces the list of assignments of a requested rank. It can contain information selected from appropriate files. The most usual data are: officer's serial number, rank, specialty, old unit, new unit, duty, and due date. This stage is also common for all ranks.

C. SYSTEM IMPLEMENTATION

The system's implementation includes all those activities that take place to convert from the old system to the new one. The new system may be totally new, replacing an existing manual or automated system, or it may be a major modification to an existing system. In either case, proper implementation adapts the system's analysis and design to provide a reliable system that meets the organization's requirements. [Seen, 1984:p. 525]

The system's implementation contains three aspects: training personnel, conversion procedures, and post-implementation review. There are methods that handle each aspect efficiently and effectively. However the problem of training personnel and conversion planning is beyond the scope of this thesis and are not discussed. As stated in the design phase, the system under development is a menu-driven system. The application follows a path of nodes through a main menu, menus, and submenus in a menu-driven format. The user is presented with a variety of choices within each type of menu node, including the ability to quit and return to the previous one.

This section of the thesis demonstrates the system implementation process through the menu-driven control mechanism. A part of the whole system is implemented to represent the assignment process. Appendix F contains listings of the appropriate programs. A partial implementation of the menu-driven portion and the assignment process, compose a representative but essential part of the whole system's implementation.

1. Initial Main Functions

This program controls the whole operation of the system and provides the root node of accessing the system.

a. Getting Started

The application system is loaded into the computer at the beginning. In fact it is not necessary for the user to know this loading process. The user must perform the following initial steps:

1. Turn on the computer.
2. Type "cd HNGS".
3. Type "dBASE".
4. Type "DO MAIN".

b. Checking Authentication

After initializing the basic dBASE III PLUS functions, the first thing that the MAIN program does is to prompt the user to insert his password in order to check its authorization to use the database system. If the user inserts an incorrect password, he is considered unauthorized and the process is aborted. The system exits automatically to the operating system (DOS), displaying a message that the user is unauthorized. Else, if the system recognizes the user as an authorized one, he is allowed to continue.

c. Main Menu

After identifying the correct authentication the MAIN program provides the user a main menu of choices as in Figure 5.1.

The functions that can be performed from the root node are: Update database, assignment process, and lists and reports, corresponding to choices 1, 2, 3 respectively. Each choice calls for an appropriate program which leads to the corresponding node menu, for further direction to the user. In addition to these three choices, the system provides two more choices: Choice "4" exit and return to the DBMS, and choice "0", quit and return to the operating system (DOS).

```
M A I N   F U N C T I O N S
---------------------------

1.  UPDATE DATABASE

2.  ASSIGNMENT PROCESS

3.  LISTS AND REPORTS

4.  EXIT AND RETURN TO DBMS

0.  QUIT AND RETURN TO DOS


ENTER YOUR SELECTION ==>_
```

Figure 5.1 Main Functions Menu

2. Second Level Functions

Three submenus direct the user to the appropriate function according to his selection of the root or main menu node.

a. Update Database

Figure 5.2 gives a sample structure of this second level functions menu. This program, UPDATEDB, controls the entire update operations.

```
U P D A T E   D A T A B A S E
-----------------------------

1.  INSERT RECORD INTO OFFICER

2.  INSERT RECORD INTO EDUTION

3.  INSEPT RECORD INTO FORLANG

4.  MODIFY RECORD INTO OFFICER

5.  MODIFY RECORD INTO FORLANG

6.  MODIFY RECORD INTO COMMAND

7.  DELETE RECORD INTO OFFICER

..............................

0.  EXIT AND RETURN TO MAIN MENU


  ENTER YOUR SELECTION ==>_
```

Figure 5.2 Update Database Menu

Through this menu it is possible for the user to insert new records into the database files, as well as to delete an existing record or to modify a set of attributes of an existing record.

Each time the user updates a record, all database files that are affected by this record are updated immediately, and automatically, without any user action.

A good alternative to the update data process is to direct through the update menu into three submenu on the third level. Each third level submenu will contain separately all the corresponding insert, delete, and modify database functions.

b. Assignment Process

Figure 5.3 illustrates the options of this second level submenu. The corresponding program, ASSPRO, controls the entire assignment processing by selecting the appropriate number.

The assignment process performs the assignments of the officers for the ranks of Ensign, First Lieutenant, Lieutenant, Lt Commander, and Commander, since there is not higher level of ranks in the warships. It is also possible to perform and the assignments of the rest of the ranks, since the system includes all the appropriate data. However, in this case, there are some other criteria which count in a different way for these ranks but have not been discussed in this thesis.

```
A S S I G N M E N T   P R O C E S S
-----------------------------------

 1.   ENSIGN

 2.   1ST LIEUTENANT

 3.   LIEUTENANT

 4.   LT COMMANDER

 5.   COMMANDER

 ................................

 0.   EXIT AND RETURN TO MAIN MENU


 ENTER YOUR SELECTION ==>_
```

Figure 5.3 Assignment Process Menu

As it is stated, the assignments are scheduled for each rank. This is done because processing each rank one at a time is simpler and easier to manage than processing all the ranks together at the same time.

The assignment process follows the assignment model design and the assignment model strategy, which are described in a separate section because of their complexity and their importance. Appendix E describes the most appropriate algorithms of the assignment process. Appendix F contains programs for the assignment of the rank 1st Lieutenant.

c. **Lists and Reports**

The LISREP program controls the entire lists and reports operations. Figure 5.3 illustrates the most useful lists and reports of the system, which also have been referred to the system analysis phase. A variety of other lists and reports can be provided by the system depending on the staff's requirements.

```
                L I S T S   A N D   R E P O R T S
                ---------------------------------

        1.  LIST OF ASSIGNMENTS OF A REQUESTED RANK

        2.  LIST OF OFFICERS OF A REQUESTED UNIT

        3.  LIST OF OFFICERS OF A REQUESTED DUTY

        4.  LIST OF OFFICERS OF A REQUESTED RANK

        5.  LIST OF OFFICERS IN A REQUESTED ORDER

        6.  REPORT OF OFFICER'S CAREER HISTORY

        7.  REPORT OF OFFICER'S CURRENT STATUS

        ............................................

        0.  EXIT TO MAIN MENU


                  ENTER YOUR SELECTION ==>_
```

Figure 5.4 Lists and Reports Menu

A good alternative to the lists and reports process is to direct through the lists and reports menu into two submenu on the third level. Each third level submenu will contain separately all the corresponding lists and reports functions.

D. SUMMARY

The model development and implementation adapt the system's analysis and design to provide a reliable system that meets the organization's requirements.

The assignment model design develops a real system model which can exist by itself. It is based on real situations, as in the military management functions thus making the process very complex.

The model has three states: Initial state "A", middle state "B", and final state "C". The assignment process takes place during the balance transformation process from the unbalanced state "B" to the balanced state "C". The model obeys the basic rules of minimum movements, the military hierarchy rule, and the rule of equivalent levels.

The assignment model strategy forms the set of all the possible functions that exist in the assignment process and affects its strategy. The whole process consists of major phases: The selecting data phase, the processing data phase, the checking data phase, and the providing result phase.

Appendix E illustrates the algorithms of the assignment process.

The system developed is a menu-driven system. The application follows a path of nodes through a main menu and submenus. The chapter demonstrates the system implementation through the menu-driven mechanism.

The initial main functions allows an authorized user to access the main functions of the system from the main menu. The second level functions direct the user to the appropriate submenu according to his selection of the root or main menu node. A part of the whole system is implemented to represent the assignment process. Appendix F contains the appropriate programs.

VI. CONCLUSIONS AND RECOMMENDATIONS

This chapter presents the conclusions and recommendations which can be logically drawn, from the research of this thesis.

A. CONCLUSIONS

The Naval Officers Personnel System is a very complex system. The existing method of officer career management is neither effective nor efficient in supporting the decision makers. The complexity in the processes causes difficulty when managing officers inside the Fleet Command.

This thesis has developed a personnel database system suitable for implementation of managing officers within the Fleet Command of the Hellenic Navy General Staff.

The main goal is to increase the productivity, effectiveness, and flexibility of the staff function as far as personnel management is concerned. This may release manpower for other purposes. Additionally, the system must support the chief and his staff in the organization of personnel, making decisions, controls, and reports with timely, relevant, up-to-date, and accurate information.

This thesis research has focused on the most complex personnel administration function, that is, the assignment process. The proposed solution will help decision makers in scheduling the annual assignments of officers to warships, as well as processing the statistical information of an officer's career.

The thesis proposes to use a computer based information processing system to maintain and analyze information related to the officers' and units' organization, and to provide information to the chief and his staff in order to support their decision making process. The thesis presents a decision support database system for the Naval Officers Management System.

To accomplish this task the following was combined: the developed organization pyramids of the Fleet Command and its operation steps, the naval officers' administration life cycle, the established model design based on scientific theory and certain military criteria and rules, the structured system analysis and design methodology of the database development process and the programming techniques and implementation. The system is designed not only according to these, but also looks beyond for possible extensions.

The developed system is considered a prototype model. Furthermore, because of the unclassified nature of this research project, some details describing the problem have been omitted and some have been considered figurative. The problem is described according to standards used by most nations.

The system implementation portion of the thesis includes only a part of the whole system design. The system is "menu-driven." This approach was chosen to provide a simple and user-friendly environment so that the users of the system can perform their jobs easier.

The software life cycle was taken into account during the program development process. Programs are so written that they would be easy to modify to meet future improvement needs.

dBASE III PLUS was used as an example of the database management system (DBMS) which could support the design and implementation of the proposed system. It is a popular relational database management system and includes both a data manipulation language and a programing language offering a host of ways to manage information.

B. RECOMMENDATIONS

As noted above, the system constructed in the thesis is a prototype model. One of the system characteristics is its ease of modification for further improvements. Improvements of the system can be done in close cooperation with the staff, which is the intended user.

The system implementation includes the most fundamental part of the design. It forms the base and the skeleton for further implementation and is representative of the features of the whole system. Using this part as a guideline, and applying the assignment processing strategy for each rank, the remaining system implementation will be routine.

All queries which may be made by personnel managers cannot be foreseen because different managers request different information. In this research, the lists and reports that are provided in the system analysis are those most usually needed in the Hellenic navy according to the author's experience. So, they are not restricted in the building of the system but serve as the first basic hints and guidelines for a more complete design. They can be modified easily according to the staff requirements

The design has included information only for officer assignments inside the Fleet Command and a certain amount of data. Future improvement could include all officers of the Hellenic Navy General Staff.

Another useful improvement could include all military personnel, that is, non-commissioned officers and seamen. More information such as address data, birthday data, body characteristics, medical information, etc., could be added to the record for each individual.

With the latest microcomputer performance and capabilities in networking this system could be put on all personnel staff offices, connected to each other via a distributed network system.

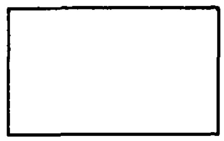
As a general conclusion, this thesis provides guidelines and constitutes a basis for future application improvements, especially in the assignment process. In addition, such a system can be used advantageously in any personnel administration function, to cover the entire area of the naval personnel management.

APPENDIX A

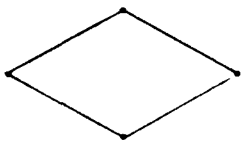
ENTITY-RELATIONSHIP (E-R) DIAGRAM

The entity-relationship (E-R) diagram shows the entity sets occurrences and their relationships. It consists of the following components:

1. Rectangles. Represent entity sets.



2. Diamonds. Represent relationships among entity sets.



3. Lines. Link entity sets to relationships.

4. Characters 1, M, N. Represent in pairs the category of a relationship.

5. Dots inside a stripe. Means that the entity's membership class is obligatory. Otherwise is non-obligatory.

6. Arrow heads are not needed.

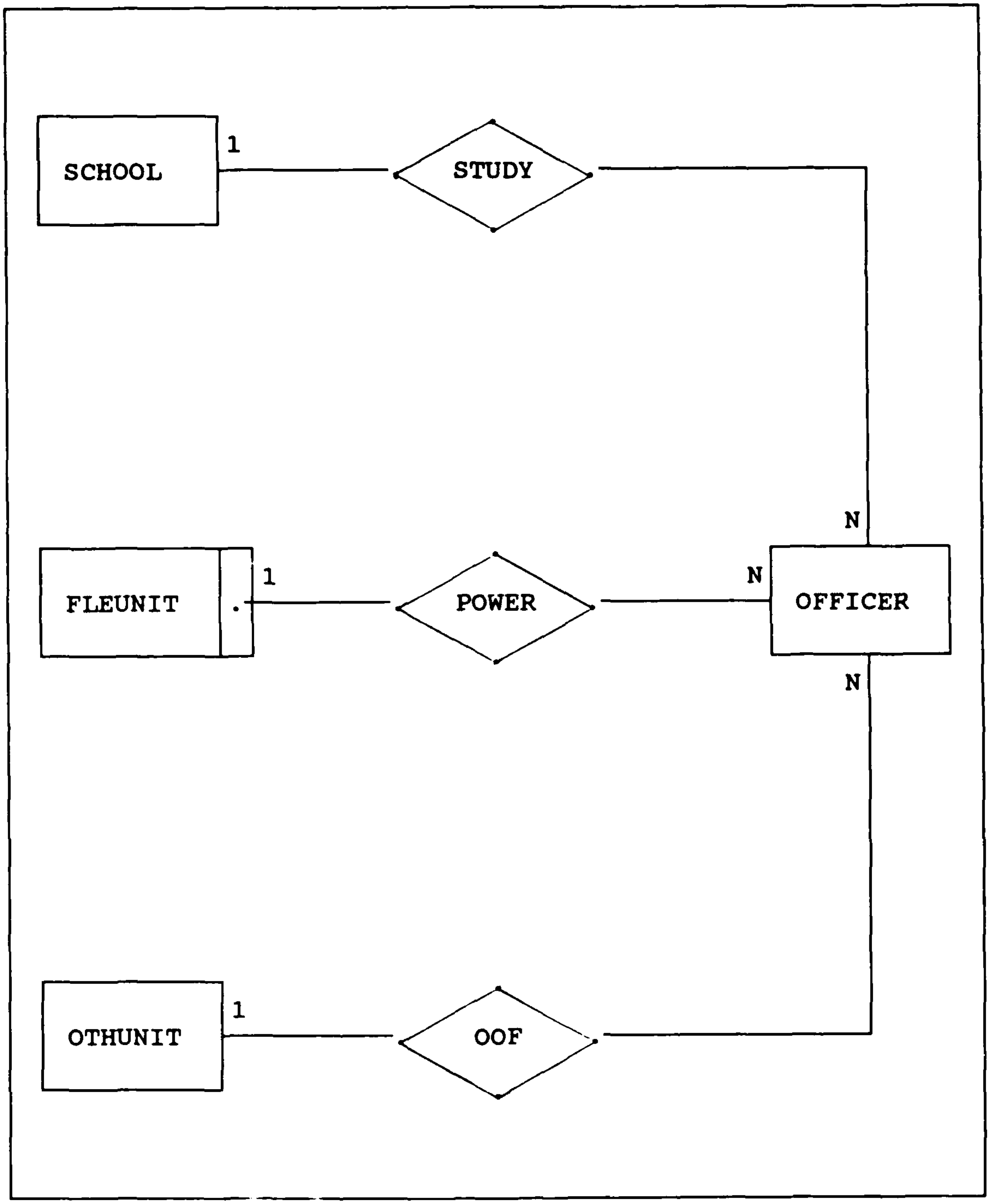


Figure A.1 Entity-Relationship (E-R) Diagram

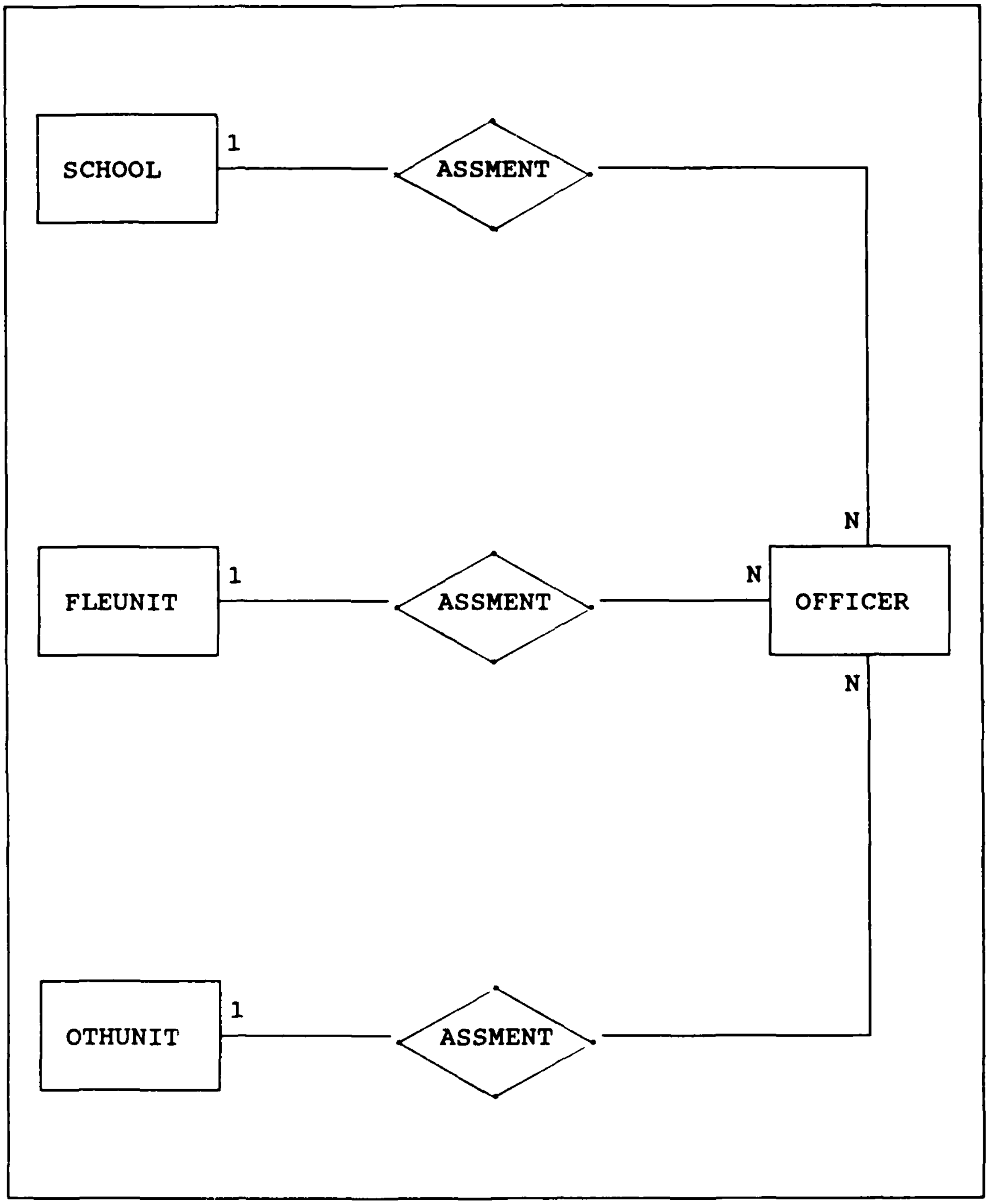


Figure A.1 (Continued)

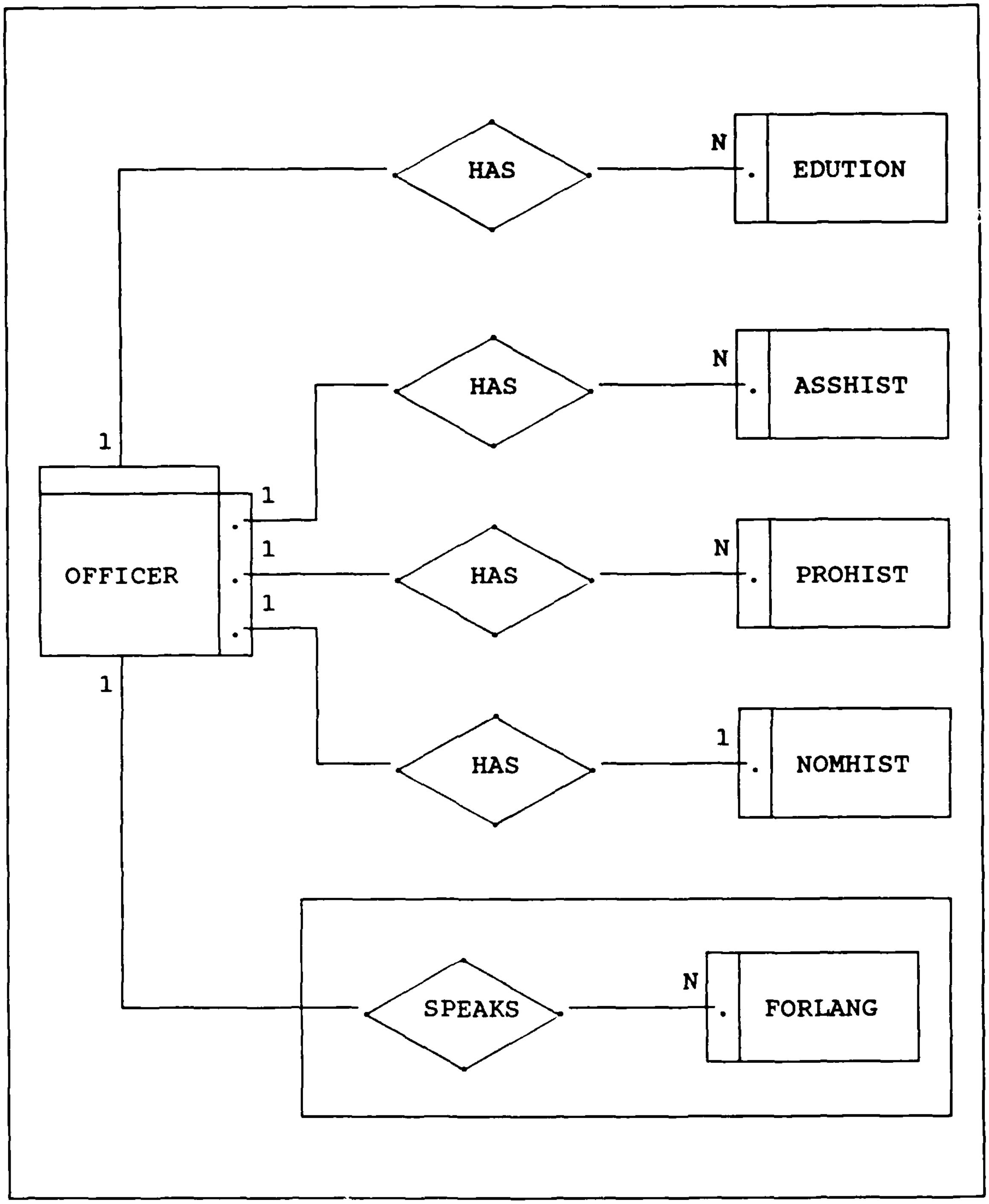


Figure A.1 (Continued)

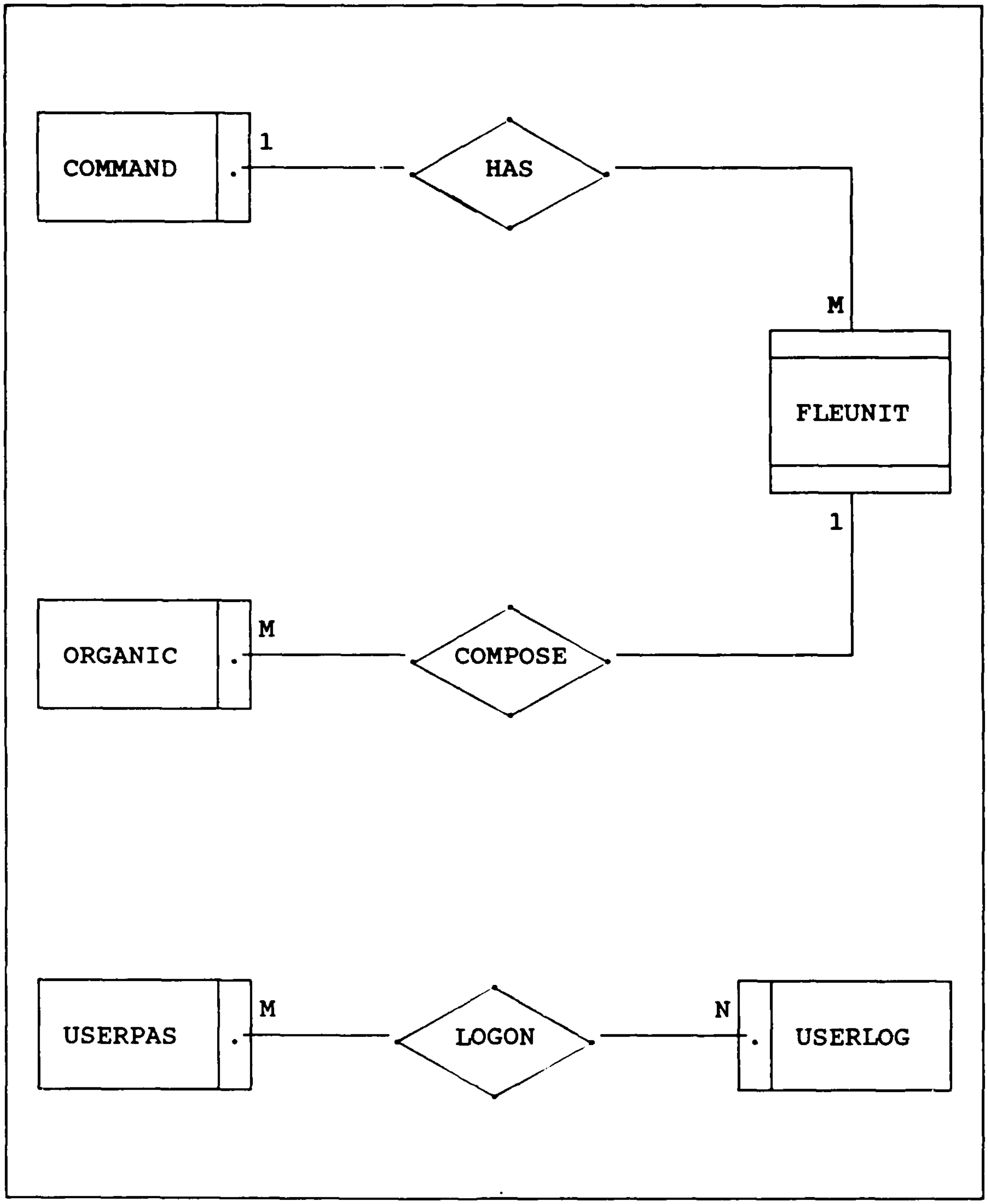


Figure A.1 (Continued)

APPENDIX B

DATABASE RELATION PRODUCTION

The database relation production comes from the process of transforming the entity-relationship (E-R) diagram into relations according to relational theory. There are two kinds of relations: The entity set relations and the relationship relations. The components of this process are the following:

1. Rectangles, diamonds, lines, characters, dots. They are same as in the entity-relationship diagram.

2. Attributes. They are properties of each relation and are written below the corresponding rectangle or diamond.

3. Asterisks (*). Represent the key attributes of a relation.

4. Consecutive dots. Means that this relation is repeated and only the key attributes is referred before the dots. The most repeated relation is the OFFICER relation.

   ....

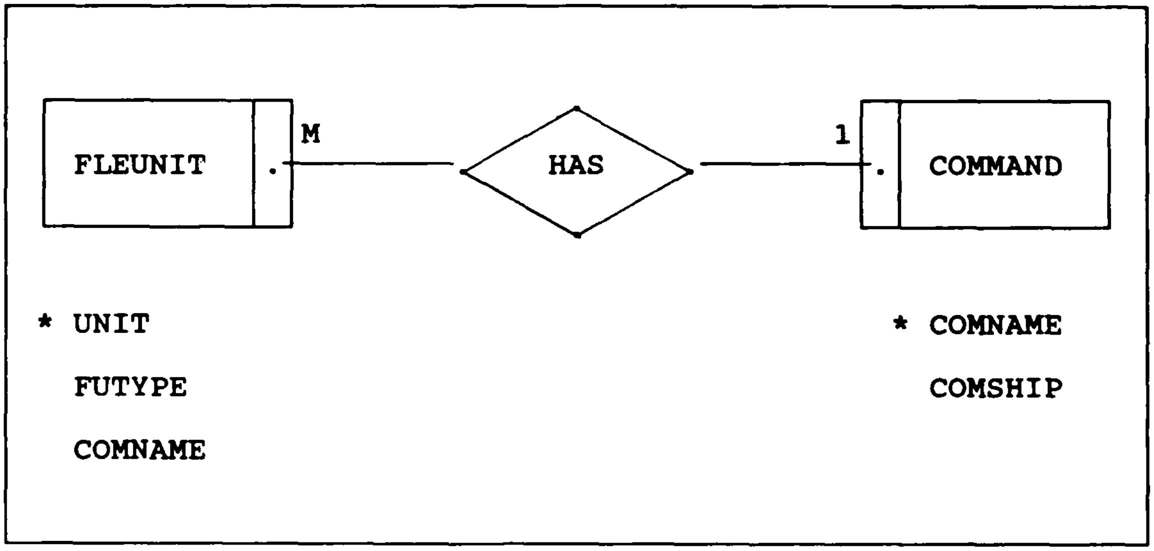


Figure B.1 Database Relation Production

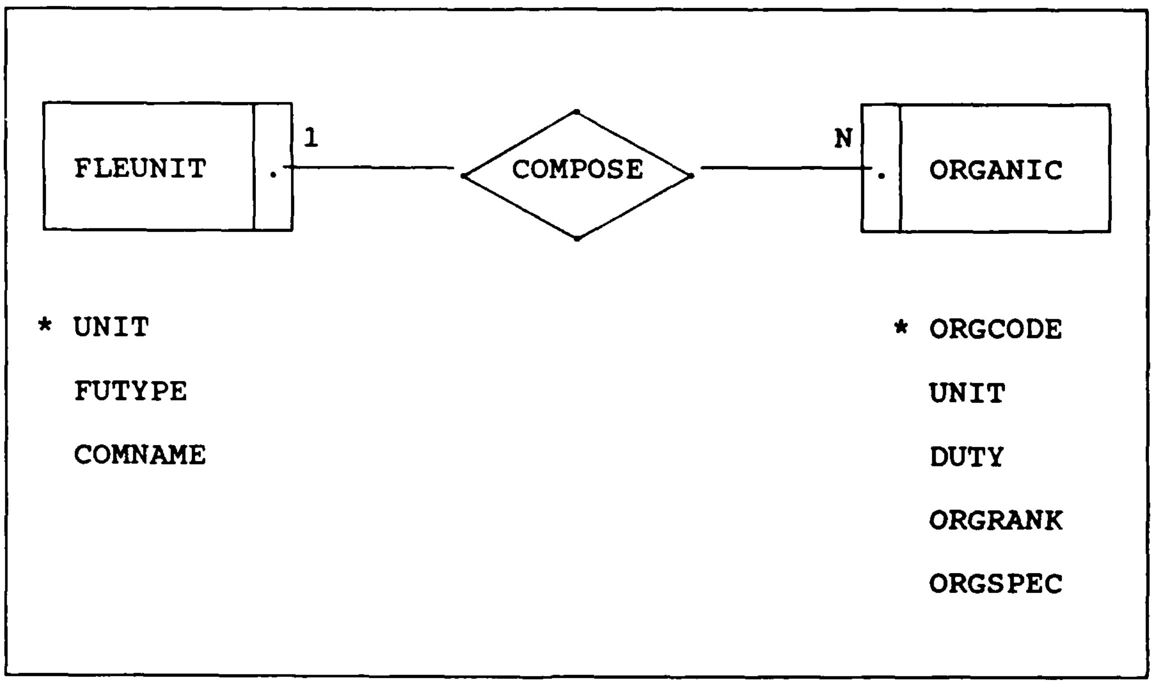


Figure B.1 (Continued)

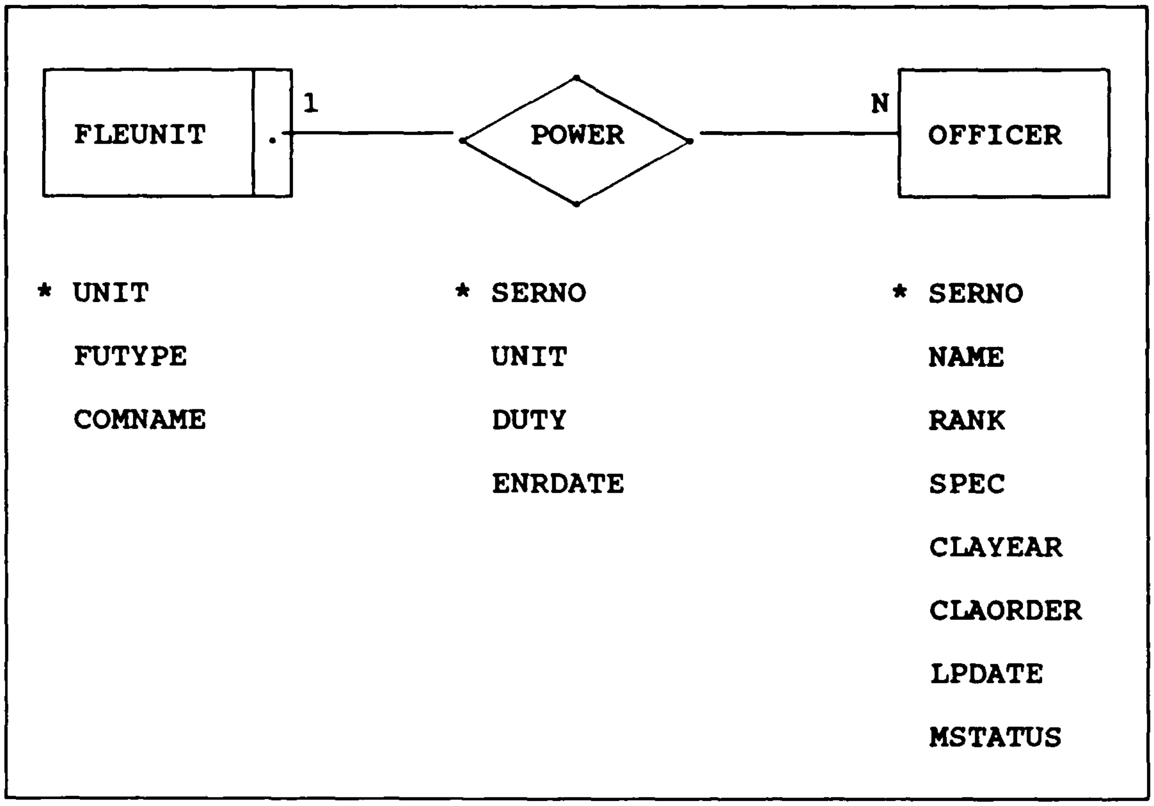


Figure B.1 (Continued)

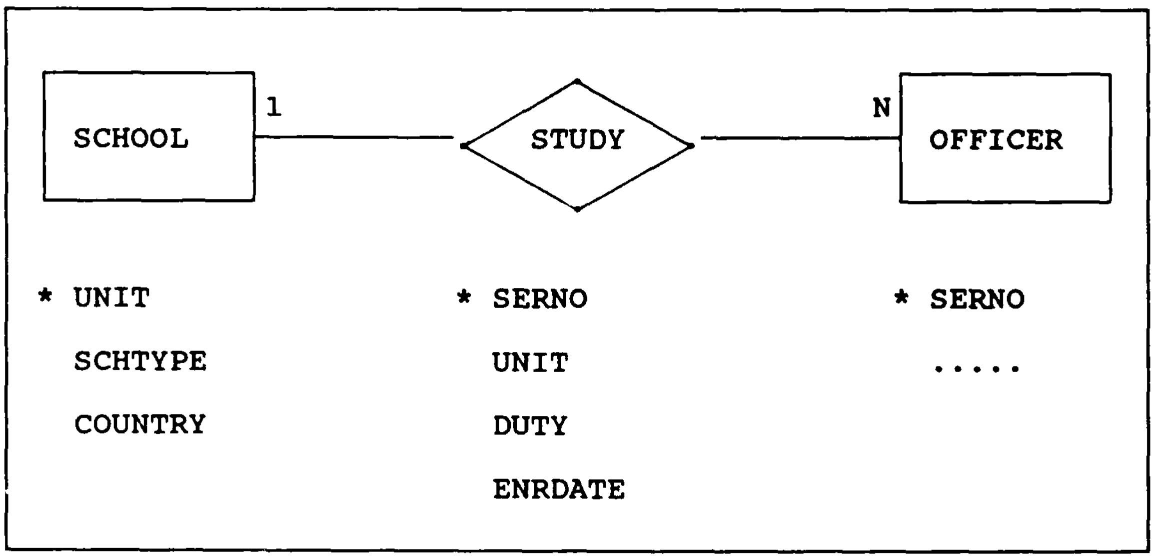


Figure B.1 (Continued)

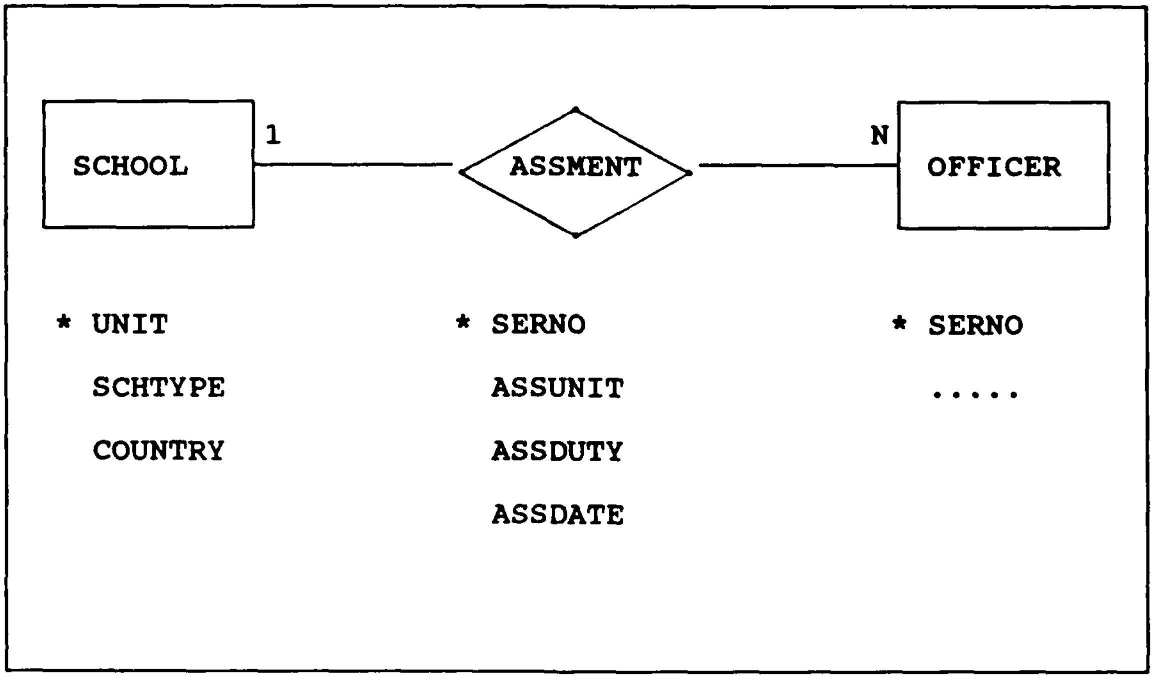


Figure B.1 (Continued)

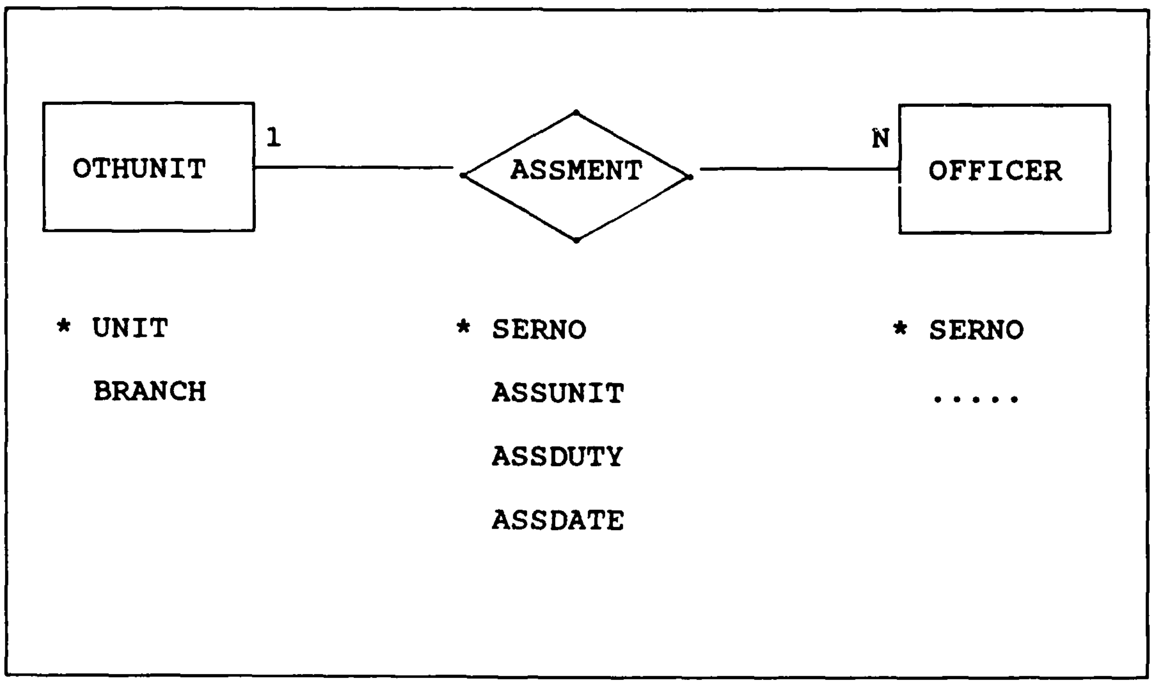


Figure B.1 (Continued)

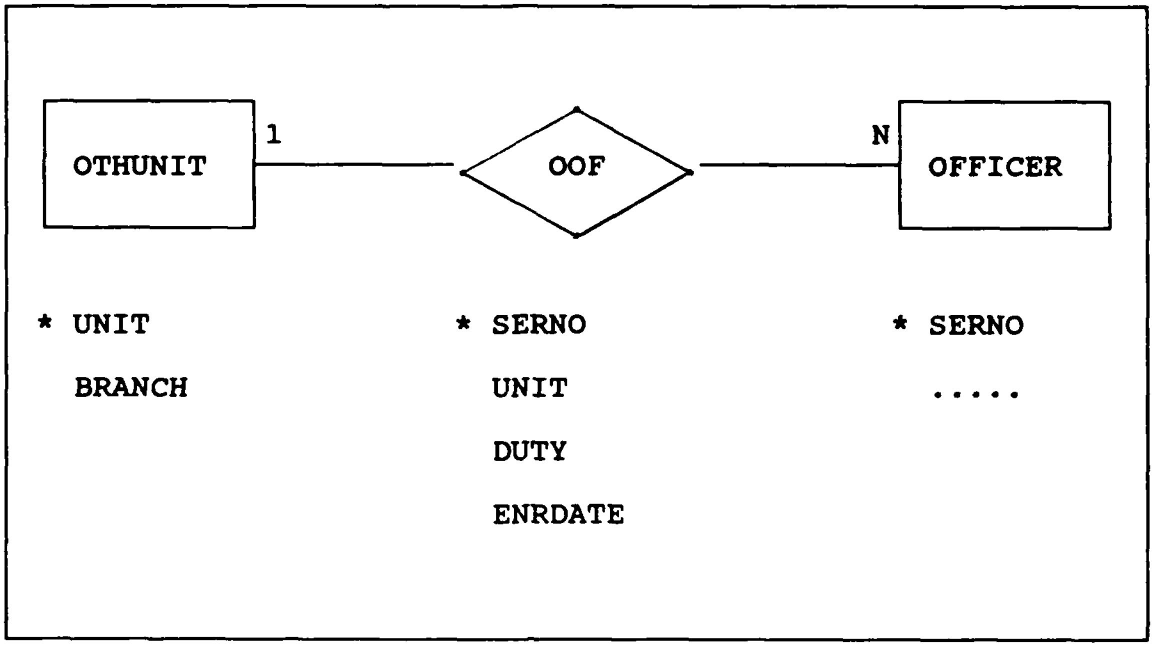


Figure B.1 (Continued)

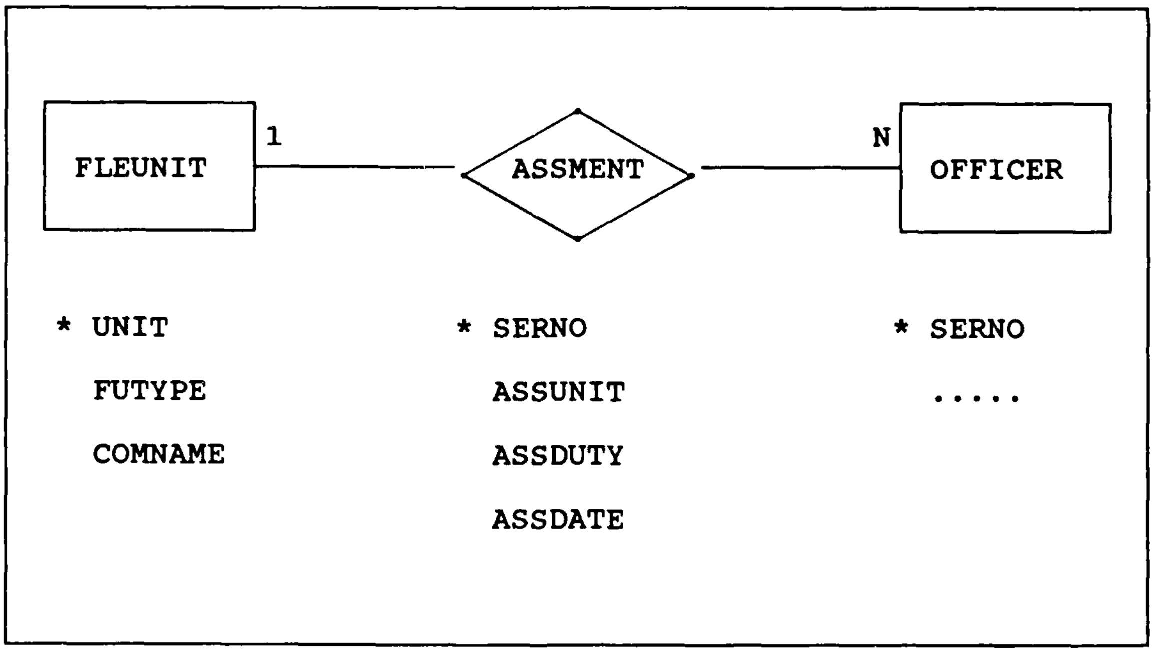


Figure B.1 (Continued)

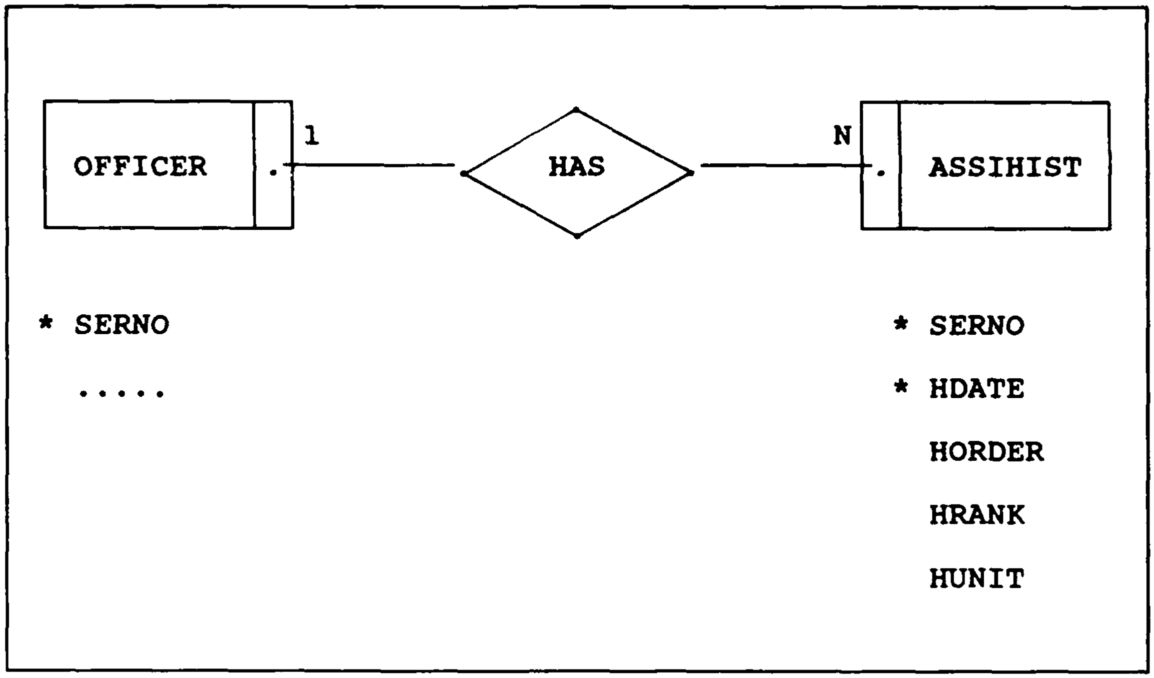


Figure B.1 (Continued)

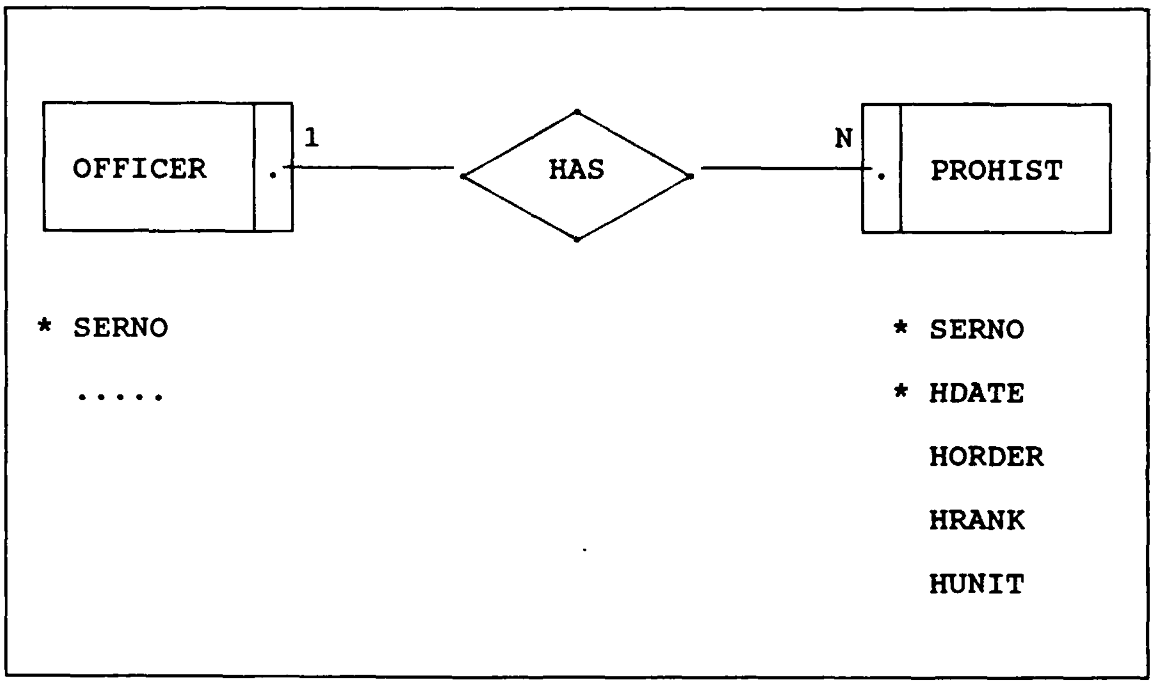


Figure B.1 (Continued)

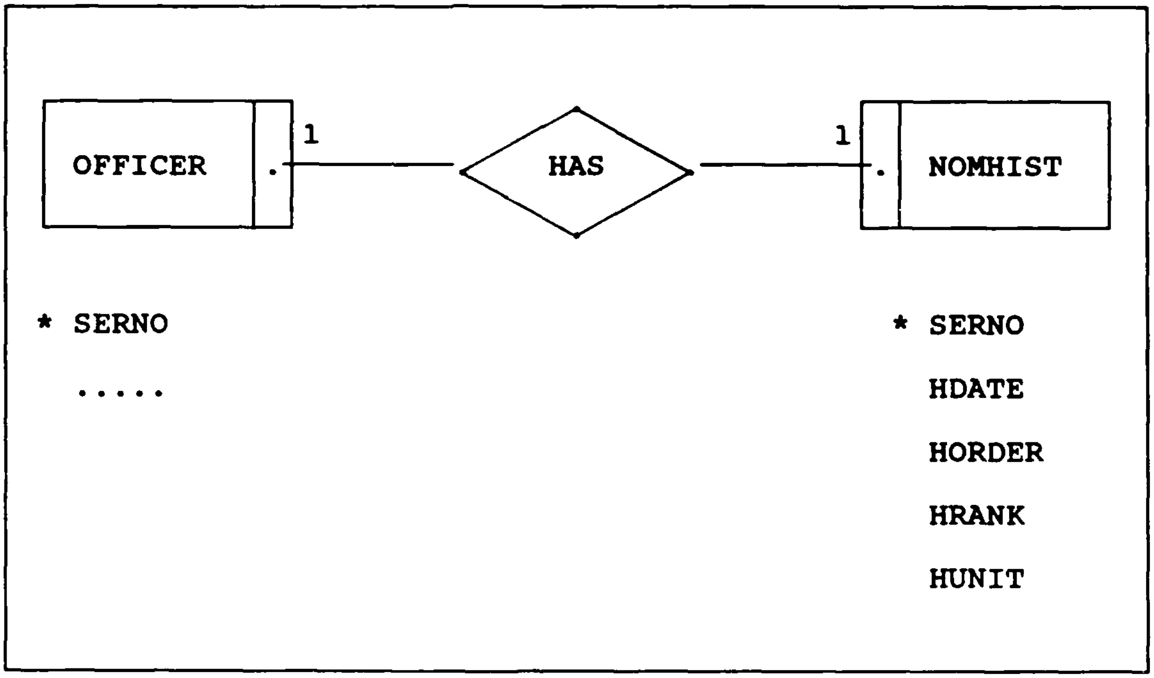


Figure B.1 (Continued)

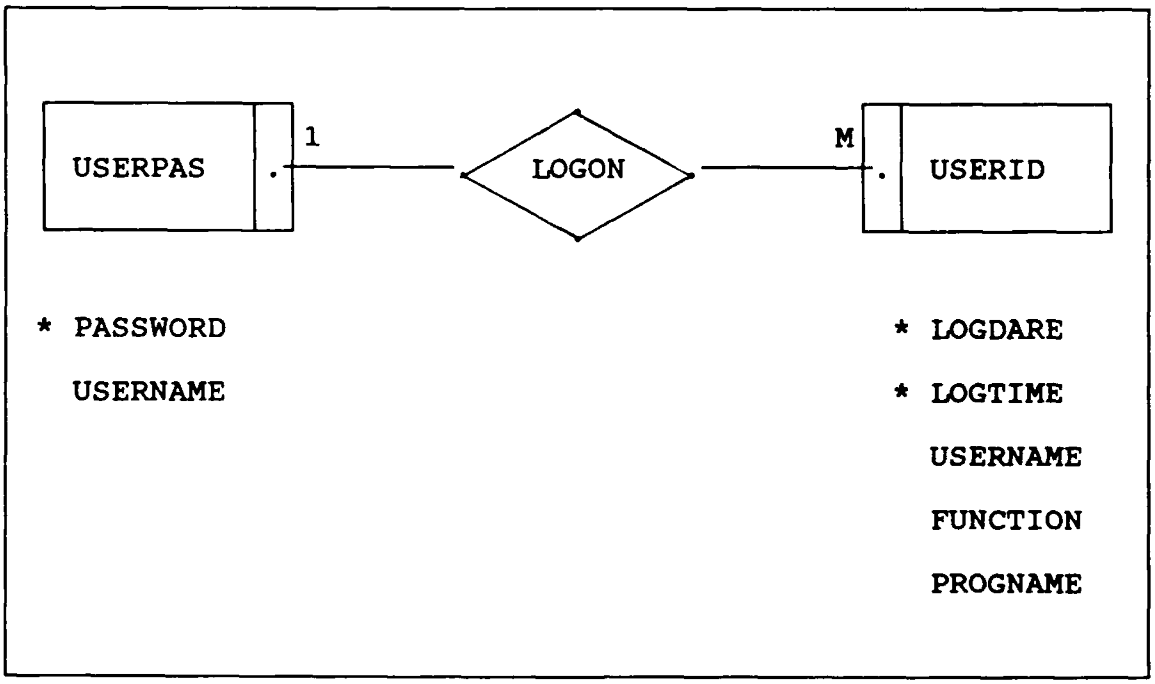


Figure B.1 (Continued)

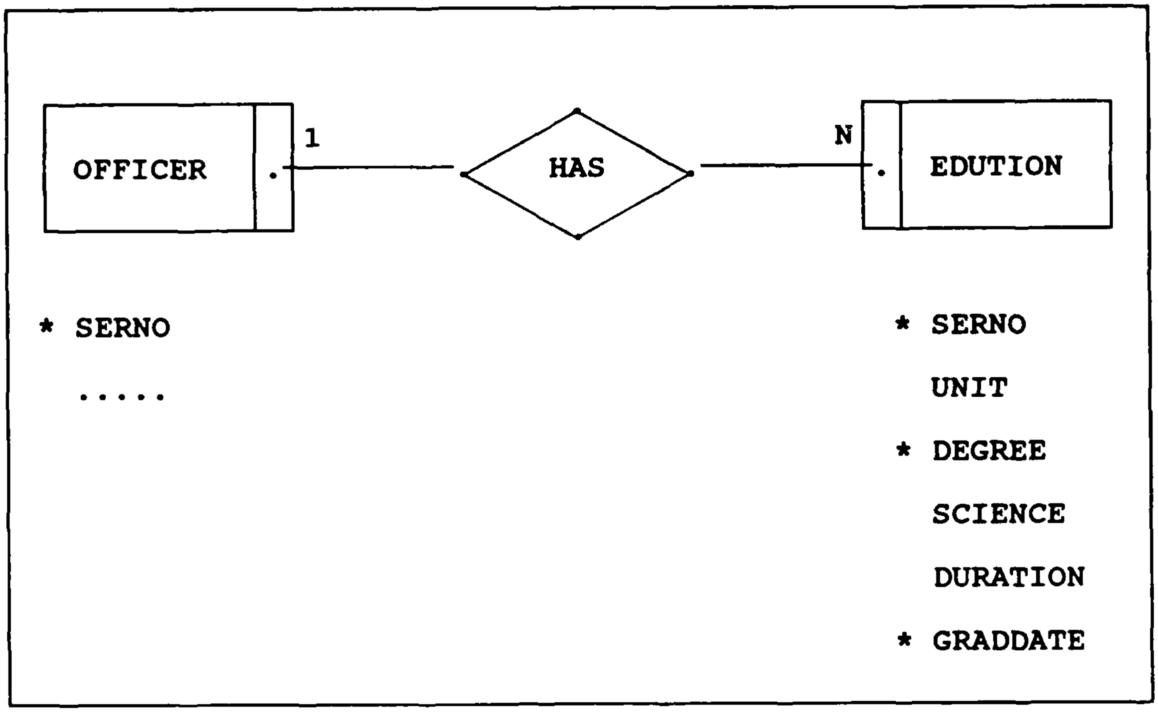


Figure B.1 (Continued)

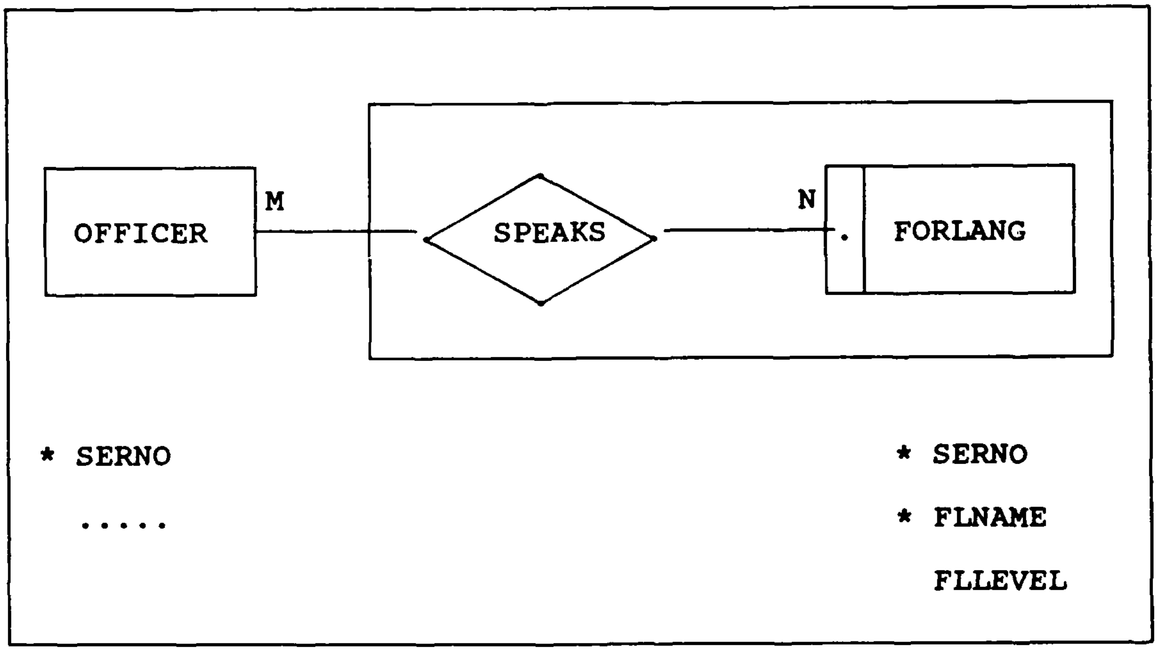


Figure B.1 (Continued)

APPENDIX C

DATABASE FUNCTIONAL DEPENDENCY

1. Relation Scheme:

    OFFICER (SERNO, NAME, RANK, SPEC, CLAYEAR
    CLAORDER, LPDATE, MSTATUS)

    Primary Key:

    SERNO

    Functional Dependency:

    CLAYEAR --> RANK

2. Relation Scheme:

    FLEUNIT (UNIT, FUTYPE, COMNAME)

    Primary Key:

    UNIT

3. Relation Scheme:

    COMMAND (COMNAME, COMSHIP)

    Primary Key:

    COMNAME

Figure C.1 Database Functional Dependency

4. <u>Relation Scheme</u>:

ORGANIC (ORGCODE, UNIT, DUTY, ORGSPEC, ORGRANK)

Primary Key:

ORGCODE

Functional Dependency:

DUTY --> ORGSPEC

(UNIT, DUTY) --> ORGRANK

5. <u>Relation Scheme</u>:

SCHOOL (UNIT, SCHTYPE, COUNTRY)

Primary Key:

UNIT

6. <u>Relation Scheme</u>:

OTHUNIT (UNIT, BRANCH)

Primary Key:

UNIT

7. <u>Relation Scheme</u>:

EDUTION (SERNO, UNIT, DEGREE, SCIENCE
DURATION, GRADATE)

Primary Key:

(SERNO, DEGREE, GRADATE)

Figure C.1 (Continued)

8. <u>Relation Scheme</u>:

ASSHIST (SERNO, HDATE, HORDER, HRANK, HUNIT)

Primary Key:

(SERNO, HDATE)

Candidate Key:

(SERNO, HORDER)

9. <u>Relation Scheme</u>:

PROHIST (SERNO, HDATE, HORDER, HRANK, HUNIT)

Primary Key:

(SERNO, HDATE)

Candidare Key:

(SERNO, HORDER)

10. <u>Relation Sheme</u>:

NOMHIST (SERNO, HDATE, HORDER, HRANK, HUNIT)

Primary Key:

SERNO

Figure C.1 (Continued)

11. <u>Relation Scheme</u>:

USERPAS (PASSWORD, USERNAME)

Primary Key:

PASSWORD

Candidate Key:

USERNAME

12. <u>Relation Scheme</u>:

USERLOG (LOGDATE, LOGTIME, USERNAME, FUNCTION PROGNAME)

Primary Key:

(LOGDATE, LOGTIME)

13. <u>Relation Scheme</u>:

POWER (SERNO, UNIT, DUTY, ENRDATE)

Primary Key:

SERNO

14. <u>Relation Scheme</u>:

STUDY (SERNO, UNIT, DUTY, ENRDATE)

Primary Key:

SERNO

Figure C.1 (Continued)

15. <u>Relation Scheme</u>:

OOF (SERNO, UNIT, DUTY, ENRDATE)

Primary Key:

SERNO

16. <u>Relation Scheme</u>:

ASSMENT (SERNO, ASSUNIT, ASSDUTY, ASSDATE)

Primary Key:

SERNO

17. <u>Relation Scheme</u>:

FORLANG (SERNO, FLNAME, FLLEVEL)

Primary Key:

(SERNO, FLNAME)

Figure C.1 (Continued)

APPENDIX D

DATABASE DICTIONARY

A. RELATION ATTRIBUTE STRUCTURE

1. Relation Officer

Key: SERNO

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	SERNO	DSERNO	Officer's serial number
02	NAME	DNAME	The name of the Officer
03	RANK	DRANK	Current Officer's rank
04	SPEC	DSPECIALTY	Officer's specialty
05	CLAYEAR	DYEAR	Class year
06	CLAORDER	DORDER	Class Order
07	LPDATE	DDATE	Last promotion date
08	MSTATUS	DMSTATUS	Marital status

2. Relation Fleunit (Fleet Unit)

Key: UNIT

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	UNIT	DUNIT	Fleet unit name
02	FUTYPE	DFUTYPE	Fleet unit type
03	COMNANE	DCOMNAME	Command name

3. Relation Command

Key: COMNAME

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	COMNAME	DCOMNAME	Command name
02	COMSHIP	DSHIP	Command ship

4. Relation Organic

Key: ORGCODE

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	ORGCODE	DORGCODE	Organic code
02	UNIT	DUNIT	Fleet unit name
03	DUTY	DDUTY	Organic duty
04	ORGSPEC	DSPECIALTY	Organic requested specialty
05	ORGRANK	DRANK	Organic requested rank

5. Relation School

Key: UNIT

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	UNIT	DUNIT	Unit school name
02	SCHTYPE	DSCHTYPE	School type
03	COUNTRY	DCOUNTRY	Country of school

6. **Relation Othunit (Other unit)**

Key: UNIT

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	UNIT	DUNIT	Other unit's name
02	BRANCH	DBRANCH	Branch

7. **Relation Eduction (Education)**

Key: (SERNO, DEGREE, GRADATE)

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	SERNO	DSERNO	Officer's serial number
02	SCHNAME	DSCHNAME	School name
03	DEGREE	DDEGREE	Degree of education
04	SCIENCE	DSCIENCE	Science of education
05	DURATION	DDURATION	Duration in months
06	GRADATE	DDATE	Graduation date

8. **Relation Asshist (Assigment History)**

Key: (SERNO, HDATE)

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	SERNO	DSERNO	Officer's serial number
02	HDATE	DDATE	Assignment history date
03	HORDER	DODRERID	Assignment history order
04	HRANK	DRANK	Assignment history rank
05	HUNIT	DUNIT	Assignment history unit

9. Relation Prohist (Promotion History)

Key: (SERNO, HDATE)

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	SERNO	DSERNO	Officer's serial number
02	HDATE	DDATE	Promotion history date
03	HORDER	DORDERID	Promotion history order
04	HRANK	DRANK	Promotion history rank
05	HUNIT	DUNIT	Promotion history unit

10. Relation Nomhist (Nomination History)

Key: SERNO

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	SERNO	DSERNO	Officer's serial number
02	HDATE	DDATE	Nomination history date
03	HORDER	DORDERID	Nomination history order
04	HRANK	DRANK	Nomination history rank
05	HUNIT	DUNIT	Nomination history unit

11. Relation Userpass (User Password)

Key: PASSWORD

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	PASSWORD	DPASSWORD	User's password
02	USERNAME	DNAME	User's name

12. Relation Userlog (User Logon)

Key: (LOGDATE, LOGTIME)

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	LOGDATE	DDATE	Date using the system
02	LOGTIME	DTIME	Time using the system
03	USERNAME	DNAME	User's name
04	FUNCTION	DFUNCTION	Function performing
05	PROGNAME	DPROGNAME	Program name executed

13. Relation Power

Key: SERNO

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	SERNO	DDATE	Officer's serial number
02	DUTY	DDUTY	Officer's duty
03	UNIT	DUNIT	Fleet unit name
04	ENRDATE	DDATE	Enrollment date

14. Relation Study

Key: SERNO

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	SERNO	DSERNO	Officer's serial number
02	DUTY	DDUTY	Officer's duty
03	UNIT	DUNIT	School unit name
04	ENRDATE	DDATE	Enrollment date

15. Relation Oof (Out of Fleet)

Key: SERNO

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	SERNO	DSERNO	Officer's serial number
02	DUTY	DDUTY	Officer's duty
03	UNIT	DUNIT	Othet unit name
04	ENRDATE	DDATE	Enrollment date

16. Relation Assment (Assignment)

Key: SERNO

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	SERNO	DSERNO	Officer's serial number
03	ASSUNIT	DUNIT	Assignment unit
04	ASSDUTY	DDUTY	Assignment duty
04	ASSDATE	DDATE	Assignment due date until the

17. Relation Forlang (Foreign Languages)

Key: (SERNO, FLNAME)

SN	ATTRIBUTE	DOMAIN	DESCRIPTION
01	SERNO	DSERNO	Officer's serial number
02	FLNAME	DFLNAME	Foreign language name
02	FLLEVEL	DFLLEVEL	Foreign language name

B. DOMAIN DEFINITION

1. Dserno

All Officers' serial numbers. A combination of five numeric character.

Type : C

Width: 5

2. Dname

The names of the officers or of any other persons who are envoled in the database system. The complete name consists of Last, First, and Middle.

Type : C

Width: 18

3. Drank

The codes which represent the ranks in the Hellenic Navy. See domain value dvrank.

Type : C

Width: 1

4. Dspecialty

The codes that represent the specialties in the Hellenic Navy. See domain value dvspecialty.

Type : C

Width: 1

5. **Dyear**

The four digits of the year.

Type : C

Width: 4

6. **Dclaorder**

The order of the Officer in his class. Each specialty has its own class. The range depends on the class size. The thesis uses a figurative domain value in [01...99].

Type : C

Width: 2

7. **Ddate**

The numeric date in format MM/DD/YY. In dBASE III PLUS exists type DATE.

Type : C

Width: 8

8. **Dmstatus**

The codes of Officer's marital status. See domain value dvmstatus.

Type : C

Width: 1

9. **Dfleunit**

The name of the Fleet unit. There are two types of units: The ships and the staffs. The domain is defined as [(DSHIP) U (DSTAFF)].

Type : C

Width: 10

10. **Dfutype**

The code that represents the type of the Fleet unit. Domain value in [1,2]. See domain value dvfutype.

Type : C

Width: 1

11. **Dcomname**

The Commands in which the Naval Fleet is organized. See domain value dvcomname.

Type : C

Width: 3

12. **Dorgcode**

The code which identifies each organic position. It consists of a combination of characters. Domain value is not provided.

Type : C

Width: 6

13. **Dduty**

The set of all the possible duties which an Officer can have during his career in the Fleet. See domain value dvduty. Only a subset of values is provided.

Type : C

Width: 4

14. **Dschool**

The schools which affect the assignments during the promotion process. They are also treated as units. Only a sample of domain values is provided (See domain value dvchool). It is a subset of the domain dunit and has the same structure.

Type : C

Width: 10

15. **Dschtype**

The type of the school. Domain value in [M, C]. See domain value dvschtype.

Type: C

Width: 1

16. **Dcountry**

The set of all the countries in the world.

Type : C

Width: 15

17. **Dothunit**

The set of the other units in the navy which do not belong to Ships, Staffs, and Schools units. They are not provided. They are a subset of the dunit domain, and have the same structure.

Type: C

Width: 10

18. **Dbranch**

The other branches of the navy except the branch of the Fleet Command. See domain value dvbranch.

Type : C

Width: 3

19. **Ddegree**

The code that represents the possible degrees of the education. See domain value dvdegree.

Type : C

Width: 1

20. **Dscience**

The science of the education. Only a sample is provided. See domain value dvscience.

Type : C

Width: 10

21. **Dduration**

Duration of education in months.

Type : C

Width: 2

22. **Dorderid**

The identification number of each issued order. It is a code of characters.

Type : C

Width: 18

23. **Dunit**

All the navy units in which officers can be assigned. It consists of the ships, staffs, schools and other units. That is: [(DSHIP) U (DSTAFF) U (DSCHOOL) U (DOTHER)].

Type : C

Width: 10

24. **Dtime**

The time in format HH:SS

Where: HH in [00 ... 23] and SS in [00 ... 59]

Type : C

Width: 5

25. **Dfunction**

The functions which the system performs. For example:

[INSERT, DELETE, MODIFY, ASSIGNMENT, LIST, REPORT, ...]

Type : C

Width: 10

26. **Dprogram**

The names of all the executed prorgams which compose the implementation of the database system.

Type : C

Width: 10

27. **Dflname**

All the foreign languages.

Type : C

Width: 12

28. **Dpassword**

Top secret code for each user of the system.

Type : C

Width: 6

29. Dship

All the warships, which belong to the Fleet Command. Subset of the dunit domain.

Type : C

Width: 10

30. Dstaff

The staffs which belong to the Fleet. Subset of the dunit domain.

Type : C

Width: 10

C. DOMAIN VALUE

1. Dvrank

CODE	DESCRIPTION
9	ENSIGN (ENS)
8	1ST LIEUTENANT (1LT)
7	LIEUTENANT (LT)
6	LT COMMANDER (LCDR)
5	COMMANDER (CDR)
4	CAPTAIN (CAPT)
3	COMMODORE (COMD)
2	REAR ADMIRAL (RADM)
1	VICE ADMIRAL (VADM)
0	ADMIRAL (ADM)

2. Dvspecialty

CODE	DESCRIPTION
D	DECK OFFICER
E	ENGINEER OFFICER
S	SUPPLY OFFICER
M	MEDICAL OFFICER
J	JUDICIAL OFFICER
C	CHEMICAL OFFICER

3. **Dvmstatus**

CODE	DESCRIPTION
M	MARRIED
U	UNMMARIED
D	DIVORCED

4. **Dvfutype**

CODE	DESCRIPTION
1	SHIP
2	STAFF

5. **Dvcomname**

CODE	DESCRIPTION
FC	FLEET COMMAND
DC	DESTROYER COMMAND
SC	SUBMARINES COMMAND
FCC	FAST CRAFT COMMAND
AC	AMPHIBIOUS COMMAND
MC	MINESWEEPERS COMMAND

6. **Dvduty**

CODE	DESCRIPTION
FCH	FLEET CHIEF
DFCH	DEPUTY FLEET CHIEF
COD	COMMANDER OFFICER
DCOD	DEPUTY COMMANDER

COFF	COMMANDING OFFICER
EXE	EXECUTIVE OFFICER
OPE	OPERATION
ADM	ADMINISTRATION
CON	COMMUNICATION
NAV	NAVIGATION
WEA	WEAPONS
ASW	ANTI-SUBMARINE WARFARE
FEC	FLEET ENGINEER CHIEF
DFEC	DEPUTY FLEET ENGINEER CHIEF
1ENG	FIRST ENGINEER
ELIC	ELECTRIC
ENIC	ELECTRONIC
DAM	DAMAGE CONTROL
ENG	MAIN ENGINES
SAN	SANITARY
SUP	SUPPLY
JUS	JUSTICE
TRAI	TRAINEE
STUD	STUDENT
NFD	NON FLEET DUTY
SPAR	SPARE

7. **Dvschool**

CODE	DESCRIPTION
NACADEMY	NAVAL OFFICERS ACADEMY
GESCHOOL	GENERAL EDUCATION SCHOOL
SESCHOOL	SPECIAL EDUCATION SCHOOL
SCOLLEGE	STAFF COLEGE
NDEFENCE	NATIONAL DEFENCE
UNIVER	UNIVERSITY

8. **Dvschtype**

CODE	DESCRIPTION
M	MILITARY SCHOOL
C	CIVILIAN SCHOOL

9. **Dvbranch**

CODE	DESCRIPTION
NLC	NAVY LOGISTIC COMMAND
NTC	NAVY TRAINING COMMAND
HGS	HEADQUORTER GENERAL STAFF

10. **Dvdegree**

CODE	DESCRIPTION
B	BACHELOR
M	MASTER
P	PHD
D	DIPLOMA

11. **Dvscience**

CODE	DESCRIPTION
MATH/TICS	MATHEMATICS
PHYSICS	PHYSICS
EENGINEER	ELECTRICAL ENGINEERING
MENGINEER	MECHANICAL ENGINEERING
CSCIENCE	COMPUTER SCIENCE
CSYSTEMS	COMPUTER SYSTEMS
WEAPONS	WEAPONS
COMM/TION	COMMUNICATION
MANAGMENT	MANAGMENT
ORESEARCH	OPERATION RESEARCH

APPENDIX E

ALGORITHMS OF ASSIGNMENT PROCESS

A. SELECT DATA

1. Asspos Data

```
CREATE file selorg1
FROM organic
WHERE orgrank = trank
JOIN selorg1 and fleunit TO orfl1
WHERE selorg1.unit = fleunit.unit
JOIN orfl1 and power TO orflpo1
WHERE orfl1.unit = power.unit .AND.
      orfl1.duty = power.duty
JOIN orflpo1 and officer TO assposx
WHERE orflpo1.serno = officer.serno
DELETE record
FROM assposx
WHERE enrdate > CTOD(lpdate) - minimum period
      .AND. YEAR(lpdate) # current year

* Check for vacant unit positions
GO TOP IN orfl1
WHILE .NOT. EOF(orfl1) DO
  GET record
  FROM orfl1
  GO TOP IN orflpo1
  FIND record
  FROM orflpo1
  WHERE record .NOT. MARK DELETED
        .AND. orflpo1.unit = orfl1.unit
        .AND. orflpo1.duty = orfl1.duty
  IF EOF(orflpo1) THEN
    * there is vacant position
    INSERT record
    FROM orfl1 INTO assposx
  ENDIF
  SKIP TO next record IN orfl1
END WHILE (orfl1)
```

2. Assoff Data

```
CREATE file seloff1
FROM officer
WHERE rank = trank
JOIN seloff1 and power TO ofpo1
```

```
WHERE seloffl.serno = power.serno
JOIN ofpol and fleunit TO assoffx
WHERE ofpol.unit = fleunit.unit
DELETE record
FROM assoffx
WHERE enrdate > CTOD(lpdate) - minimum period
      .AND. YEAR(lpdate) # current year
JOIN seloffl and study TO comel
WHERE seloffl.serno = study.serno
JOIN seloffl and oof TO come2
WHERE seloffl.serno = study.serno
APPEND comel TO assoffx
APPEND come2 TO assoffx
```

B. **PROCESS DATA**

1. **Hierarchy Levels**

```
INDEX ON rank + clayear + claorder
WITHIN assposx
INDEX ON rank + clayear + claorder
WITHIN assoffx
```

2. **Trainee Data**

```
finish = FALSE
again = TRUE
* Select all destroyers warships
CREATE file seldes
FROM fleunit
WHERE comname = "DC"
* Select all new Ensign officers
Create file selens
FROM officer
WHERE officer.clayear = current year

WHILE .NOT. finish DO
  * start new pass
  IF again = TRUE THEN
    GO TOP IN seldes
  ENDIF
  again = false
  WHILE .NOT. EOF(seldes) DO
    GET record
    FROM seldes
    okl = FALSE
    GO TOP IN selens
    FIND record
    FROM selens
    WHERE record .NOT. MARK DELETED
    IF .NOT. EOF(selens) THEN
      * find record
      GET record
      FROM assoff
```

```
        INSERT record INTO assment
        WHERE assment.serno = selens.serno
              assment.assunit = seldes.unit
              assment.assduty = "TRAINEE"
              assment.assdate = tdate
        MARK DELETED current record IN selens
        ok1 = TRUE
      ENDIF
      SKIP TO next record in seldes
      * Check ok condition
      IF ok1 = TRUE THEN
        IF EOF(seldes) THEN
          again = TRUE
          * for startig next pass
        ENDIF
      ELSE if ok1 = FALSE
        finish = TRUE
      ENDIF
    END WHILE (seldes)
  END WHILE (finish)
```

3. **Student Data**

```
  GO TOP IN assoff
  WHILE .NOT. EOF(assoff) DO
    GET record
    FROM assoff
    IF assoff.clayear = tclayear .AND.
          record .NOT. MARK DELETED THEN
    INSERT record INTO assment
    WHERE assment.serno = assoff.serno
          assment.assunit = tunit
          assment.assduty = "STUDENT"
          assment.assdate = tdate
    MARK DELETED current record IN assoff
    SKIP TO next record IN assoff
  END WHILE (assoff)
```

4. **Staff Data**

```
  done = FALSE
  GO TOP IN asspos
  WHILE .NOT. EOF(asspos) DO
    GET record
    FROM asspos
    ok1 = FALSE
    IF futype = "STAFF" .AND.
      record .NOT. MARK DELETED THEN
      GO TOP IN assoff
      FIND record
      FROM assoff
```

```
          WHERE record .NOT. MARK DELETED .AND.
                assoff. clayear = tclayear .AND.
                assoff.futype # "STAFF"
          IF .NOT. EOF(asspos) THEN
            * find record
            GET record
            FROM asspos
            INSERT record INTO assment
            WHERE assment.serno = assoff.serno
                  assment.assunit = asspos.unit
                  assment.assduty = asspos.duty
                  assment.assdate = tdate
            MARK DELETED current record IN assoff
            ok1 = TRUE
          ELSE if EOF(assoff)
            * not find record
            done = TRUE
          ENDIF
          * Check ok condition
          IF ok1 = TRUE THEN
            MARK DELETED current record IN asspos
          ENDIF
          SKIP TO next record IN asspos
        ENDIF
      END WHILE (asspos)

        * There are not other officers
      IF done = TRUE  THEN
        GO TOP IN asspos
        WHILE .NOT. EOF(asspos) DO
          GET record
          FROM asspos
          ok1 = FALSE
          IF record IN asspos .NOT. MARK DELETED
            .AND. futype = "STAFF" THEN
            GO TOP IN assoff
            FIND record
            FROM assoff
            WHERE assoff.serno = asspos.serno
                  .AND. .NOT. MARK DELETED
            IF .NOT. EOF(assoff) THEN
              * find record
              MARK DELETED current record IN assoff
              ok1 = TRUE
            ENDIF
          ENDIF
          * Check ok condition
          IF ok1 = TRUE THEN
            MARK DELETED current record IN asspos
          ENDIF
          SKIP TO next record IN asspos
```

```
    END WHILE (asspos)
  ENDIF

  * optional function
  IF done = TRUE THEN
    GO TOP IN assoff
    WHILE .NOT. EOF(assoff) DO
      GET record
      FROM assoff
      IF record IN assoff .NOT. MARK DELETED
        .AND. futype = "STAFF" THEN
        INSERT record INTO assment
        WHERE assment.serno = asspos.serno
              assment.assunit = "HGS"
              assment.assduty = "NFD"
              assment.assdate = tdate
        MARK DELETE current record IN assoff
      ENDIF
    END WHILE(assoff)
  ENDIF
```

5. Specific Data

```
  done = FALSE
  GO TOP IN asspos
  WHILE .NOT. EOF(asspos) DO
    GET record
    FROM asspos
    ok1 = FALSE
    IF duty = "tduty" .AND.
      record .NOT. MARK DELETED THEN
      GO TOP IN assoff
      FIND record
      FROM assoff
      WHERE record .NOT. MARK DELETED .AND.
            assoff. clayear = tclayear .AND.
            assoff.duty # tduty
      IF .NOT. EOF(asspos) THEN
        * find record
        GET record
        FROM asspos
        INSERT record INTO assment
        WHERE assment.serno = assoff.serno
              assment.assunit = asspos.unit
              assment.assduty = asspos.duty
              assment.assdate = tdate
        MARK DELETED current record IN assoff
        ok1 = TRUE
      ELSE if EOF(assoff)
        * not find record
        done = TRUE
      ENDIF
```

```
    * Check ok condition
    IF ok1 = TRUE THEN
      MARK DELETED current record IN asspos
    ENDIF
    SKIP TO next record IN asspos
  ENDIF
END WHILE (asspos)

  * There are not other officers
IF done = TRUE THEN
  GO TOP IN asspos
  WHILE .NOT. EOF(asspos) DO
    GET record
    FROM asspos
    ok1 = FALSE
    IF record IN asspos .NOT. MARK DELETED
      .AND. duty = tduty THEN
      GO TOP IN assoff
      FIND record
      FROM assoff
      WHERE assoff.serno = asspos.serno
            .AND. .NOT. MARK DELETED
      IF .NOT. EOF(assoff) THEN
        * find record
        MARK DELETED current record IN assoff
        ok1 = TRUE
      ENDIF
    ENDIF
    * Check ok condition
    IF ok1 = TRUE THEN
      MARK DELETED current record IN asspos

    ENDIF
    SKIP TO next record IN asspos
  END WHILE (asspos)
ENDIF

* optional function
IF done = TRUE THEN
  GO TOP IN assoff
  WHILE .NOT. EOF(assoff) DO
    GET record
    FROM assoff
    IF record IN assoff .NOT. MARK DELETED
      .AND. duty = tduty THEN
      INSERT record INTO assment
      WHERE assment.serno = asspos.serno
            assment.assunit = "HGS"
            assment.assduty = "NFD"
            assment.assdate = tdate
      MARK DELETE current record IN assoff
```

```
      ENDIF
    END WHILE(assoff)
  ENDIF
```

6. **Rule of Hierarchy**

```
  GO TOP IN asspos
  WHILE .NOT. EOF(asspos) DO
    GET record
    FROM asspos
    ok1 = FALSE
    IF record .NOT. MARK DELETED
      .AND. level-condition THEN
      GO TOP IN assoff
      FIND record
      FROM assoff
      WHERE record .NOT. MARK DELETED
      IF .NOT. EOF(assoff) THEN
        * find record
        GET record
        FROM assoff
        INSERT record INTO assment
        WHERE assment.serno = assoff.serno
              assment.assunit = asspos.unit
              assment.assduty = asspos.duty
              assment.assdate = tdate
        MARK DELETE current record IN assoff
        ok1 = TRUE
      ENDIF
    ENDIF
    * Check ok condition
    IF ok1 = TRUE THEN
      MARK DELETED current record IN asspos
    ENDIF
    SKIP TO next record IN asspos
  END WHILE
```

7. **Rule of Equivalent Levels**

```
  GO TOP IN asspos
  WHILE .NOT. EOF(asspos) DO
    GET record
    FROM asspos
    ok1 = FALSE
    IF record IN asspos .NOT. MARK DELETED
      .AND. level-condition THEN
      GO TOP IN assoff
      FIND record
      FROM assoff
      WHERE assoff.serno = asspos.serno
            .AND. equivalent-level-condition
            .AND. record .NOT. MARK DELETED
```

```
        IF .NOT. EOF(assoff) THEN
          * find record
          MARK DELETED current record IN assoff
          ok2 = TRUE
        ENDIF
      ENDIF
      * Check ok condition
      IF ok2 = TRUE THEN
        MARK DELETED current record IN asspos
      ENDIF
      SKIP TO next record IN asspos
    END WHILE (asspos)
```

8. **Rule of Minimum Movements**

```
    GO TOP IN asspos
    WHILE .NOT. EOF(asspos) DO
      GET record
      FROM asspos
      ok3 = FALSE
      IF record .NOT. MARK DELETED
        .AND level-condition1 THEN
        GO TOP IN assoff
        FIND record
        FROM assoff
        WHERE record .NOT. MARK DELETED
              .AND. assoff.unit = asspos.unit
              .AND. level-condition2
        IF .NOT. EOF(assoff) THEN
          * find record
          IF assoff.serno = asspos.serno
            MARK DELETED current record IN assoff
          ELSE
            GET record
            FROM asspos
            INSERT record INTO assment
            WHERE assment.serno = assoff.serno
                  assment.assunit = asspos.unit
                  assment.assduty = asspos.duty
                  assment.assdate = tdate
            MARK DELETE current record IN assoff
          ENDIF
          ok3 = TRUE
        ENDIF
      ENDIF
      * Check ok condition
      IF ok3 = TRUE THEN
        MARK DELETED current record IN asspos
      ENDIF
      SKIP TO next record IN asspos
    END WHILE (asspos)
```

C. CHECK DATA

```
posdone = FALSE
offdone = FALSE
GO TOP IN asspos
FIND record
FROM asspos
WHERE record IN asspos .NOT. MARK DELETED
IF EOF(asspos) THEN
  posdone = TRUE
ENDIF
GO TOP IN assoff
FIND record
FROM assoff
WHERE record IN assoff .NOT. MARK DELETED
IF EOF(assoff) THEN
  offdone = TRUE
ENDIF

IF offdone .AND. prdone THEN
  "NO AVAILABLE POSITIONS"
  "NO AVAILABLE OFFICERS"
ENDIF

IF .NOT. offdone THEN
  STORE "0" TO answer
  GO TOP IN assoff
  FIND record
  FROM assoff
  WHERE record .NOT. MARK DELETED
  WHILE .NOT. EOF(assoff) DO
    "THERE ARE STILL AVAILABLE OFFICERS"
    "1. ASSIGN IN FC"
    "2. ASSIGN IN HGS"
    IF answer = 1 THEN
      GET record
      FROM assoff
      INSERT record INTO assment
      WHERE assment.serno = assoff.serno
            assment.assunit = "FCS"
            assment.assduty = "SPARE"
            assment.assdate = tdate
      MARK DELETE current record IN assoff
    ELSE
      * answer = 2
      GET record
      FROM assoff
      INSERT record INTO assment
      WHERE assment.serno = assoff.serno
            assment.assunit = "HGS"
            assment.assduty = "NFD"
            assment.assdate = tdate
      MARK DELETED current record IN assoff
```

```
    ENDIF (answer)
  END WHILE (assoff)
ENDIF (offdone)

IF .NOT. posdone THEN
  "THERE ARE STILL AVAILABLE POSITIONS"
  GO TOP IN asspos
  WHILE .NOT. EOF(asspos) DO
    FIND record
    FROM asspos
    WHERE record .NOT. MARK DELETED
    GET record
    FROM asspos
    DISPLAY record
    MARK DELETED current record IN asspos
  END WHILE (asspos)
ENDIF (posdone)
```

D. **RESULT DATA**

```
CREATE file lst1
FROM officer
WHERE rank = reqrank
JOIN lst1 and power TO lst21
WHERE lst1.serno = power.serno
JOIN lst1 and study TO lst22
WHERE lst1.serno = study.serno
JOIN lst1 and oof TO lst23
WHERE lst1.serno = oofserno
APPEND lst22 TO lst21
APPEND lst23 TO lst21
IF reqrank = "9" THEN
  APPEND lst1 TO lst21
  WHERE clayear = current year
ENDIF
JOIN lst21 and assment TO lart
WHERE lst21.serno = assment.serno
JOIN lart and dvrank TO lar
WHERE lart.rank = dvrank.rank
INDEX ON spec + rank + clayear + claorder
WITHIN lar
DISPLAY REPORT FORM lar
```

APPENDIX F

DATABASE SYSTEM PROGRAMS

A. MAIN PROGRAMS

```
*** PROGRAM MAIN
* This is the main program, which controls the entire
* database system

CLEAR
* Initialize basic dBASE III functions
SET TALK OFF
SET DELIMITER OFF
SET HEADING OFF
SET EXACT ON

PUBLIC psw
STORE '      ' TO psw
@ 10,18 TO 12,65 DOUBLE

* Check user's authorization
@ 11,30 SAY 'ENTER PASSWORD ->'
        SET CONSOLE OFF
        ACCEPT TO psw
        SET CONSOLE ON
USE userpas
LOCATE FOR password = UPPER(psw)
IF EOF()
  SET COLOR TO W*
  @ 11,28 SAY '     UNAUTHORIZED USER      '
  DO delay
  SET COLOR TO W/B, G/R, BG
  QUIT
ENDIF

DO title
DO delay
DO delay
STORE .T. TO contmm
DO WHILE contmm
   DO mainmenu
   * perform appropriate function
```

```
   DO CASE
      CASE selmm = 0
         CLEAR
         CLEAR ALL
         QUIT
      CASE selmm = 1
         DO updatedb
      CASE selmm = 2
         DO asspro
      CASE selmm = 3
         DO lisrep
      CASE selmm = 4
         CLEAR
         STORE .F. contmm
   ENDCASE
ENDDO

SET TALK ON
SET DELIMITER ON
SET EXACT OFF
SET HEADING ON
CLEAR ALL
RETURN

* EOF: MAIN.PRG
```

```
*** PROGRAM MAINMENU
* This program desplays the main root menu

CLEAR
PUBLIC selmm
STORE 0 TO selmm
@  2,0 TO 18,79 DOUBLE
@  4,1 TO  4,78 DOUBLE
@ 16,1 TO 16,78 DOUBLE
SET COLOR TO W*
@  3,30 SAY [M A I N  F U N C T I O N S]
SET COLOR TO N/BR
@  7,30 SAY [1. UPDATE DATABASE        ]
@  8,30 SAY [2. ASSIGNMENT PROCESS     ]
@  9,30 SAY [3. LISTS AND REPORTS      ]
@ 11,30 SAY [4. EXIT AND RETURN TO DBMS]
@ 13,30 SAY [0. QUIT AND RETURN TO DOS ]
SET COLOR TO W+
@ 17,30 SAY 'ENTER YOUR SELECTION ==>' GET selmm;
        PICTURE "9" RANGE 0,4
READ
SET COLOR TO W/B, G/R, BG
RETURN

* EOF: MAINMENU.PRG
```

```
*** PROGRAM TITLE
* This program displays the title

CLEAR
@  2,0 TO 19,79 DOUBLE
@  5,20 SAY [    HELLENIC NAVY GENERAL STAFF      ]
@  7,20 SAY [          FLEET COMMAND              ]
@ 11,20 SAY [  DECISION SUPPORT DATABASE SYSTEM   ]
@ 13,20 SAY [              FOR                    ]
@ 15,20 SAY [ THE NAVAL OFFICERS MANAGMENT STAFF  ]
RETURN

* EOF: TITLE.PRG
```

B. UPDATE DATABASE PROGRAMS

```
*** PROGRAM UPDATEDB
* This program controls the update operations

SET COLOR TO W/B, G/R, BG
CLEAR
STORE .T. TO updacont
PUBLIC udcode
DO WHILE updacont
  DO udmenu
  DO CASE
    CASE udcode = 0
      STORE  .F. TO updacont
    CASE udcode = 1
      DO insoff
      STORE .T. TO updacont
    CASE udcode = 2
      DO insedu
      STORE .T. TO updacont
    CASE udcode = 3
      DO insforl
      STORE .T. TO updacont
    CASE udcode = 4
      DO modoff
      STORE .T. TO updacont
    CASE udcode = 5
      DO modforl
      STORE .T. TO updacont
    CASE udcode = 6
      DO modcom
      STORE .T. TO updacont
    CASE udcode = 7
      DO deloff
      STORE .T. TO updacont
  ENDCASE
ENDDO
RETURN

* EOF: UPDATEDB.PRG
```

```
*** PROGRAM UDMENU
* This program desplays the update database menu

CLEAR
PUBLIC udcode
STORE 0 TO udcode
@  2,0 TO 21,79 DOUBLE
@  4,1 TO  4,78 DOUBLE
@ 19,1 TO 19,78 DOUBLE
SET COLOR TO N*/BR
@  3,22 SAY [U P D A T E  D A T A B A S E]
SET COLOR TO N/BR
@  6,20 SAY [ 1. INSERT RECORD INTO OFFICER   ]
@  7,20 SAY [ 2. INSERT RECORD INTO EDUTION   ]
@  8,20 SAY [ 3. INSERT RECORD INTO FORLANG   ]
@  9,20 SAY [ 4. MODIFY RECORD INTO OFFICER   ]
@ 10,20 SAY [ 5. MODIFY RECORD INTO FORLANG   ]
@ 11,20 SAY [ 6. MODIFY RECORD INTO COMMAND   ]
@ 12,20 SAY [ 7. DELETE RECORD INTO OFFICER   ]
@ 13,20 SAY [ ............................    ]
@ 15,20 SAY [ 0. EXIT AND RETURN TO MAIN MENU ]
SET COLOR TO W+/BR
@ 20,25 SAY 'ENTER YOUR SELECTION ==>' ;
        GET udcode PICTURE "9" RANGE 0,7
READ
SET COLOR TO W/B, G/R, BG
RETURN

* EOF: UDMENU.PRG
```

C. ASSIGNMENT PROCESS PROGRAMS

```
*** PROGRAM ASSPRO
* This program controls the assignment process

SET COLOR TO W/B, G/R, BG
CLEAR
STORE .F. TO stop
PUBLIC apcode
DO WHILE .NOT. STOP
  DO apmenu
  DO CASE
    CASE apcode = 0
      RETURN
    CASE apcode = 1
      DO assign9
      STORE .F. TO stop
    CASE apcode = 2
      DO assign8
      STORE .F. TO stop
    CASE apcode = 3
      DO assign7
      STORE .F. TO stop
    CASE apcode = 4
      DO assign6
      STORE .F. TO stop
    CASE apcode = 5
      DO assign5
      STORE .F. TO stop
  ENDCASE
ENDDO
RETURN

* EOF: ASSPRO.PRG
```

```
*** PROGRAM APMENU
* This program desplays the assigment process menu

CLEAR
PUBLIC apcode
STORE 0 TO apcode
@  2,0 TO 21,79 DOUBLE
@  4,1 TO  4,78 DOUBLE
@ 19,1 TO 19,78 DOUBLE
SET COLOR TO N*/BR
@  3,20 SAY [A S S I G N M E N T  P R O C E S S]
SET COLOR TO N/BR
@  6,20 SAY [ 1. ENSIGN                         ]
@  7,20 SAY [ 2. 1ST LIEUTENANT                 ]
@  8,20 SAY [ 3. LIEUTENANT                     ]
@  9,20 SAY [ 4. LT COMMANDER                   ]
@ 10,20 SAY [ 5. COMMANDER                      ]
@ 11,20 SAY [ ............................... ]
@ 15,20 SAY [ 0. EXIT AND RETURN TO MAIN MENU ]
SET COLOR TO W+/BR
@ 20,20 SAY 'ENTER YOUR SELECTION ==>' ;
        GET apcode PICTURE "9" RANGE 0,5
READ
SET COLOR TO W/B, G/R, BG
RETURN

* EOF: APMENU.PRG
```

```
***  PROGRAM ASSIGN8
* This program performs the assignments of the
* 1st Lieutenants and updates the USERLOG file

CLEAR
@ 1,5 SAY  "ASSIGNMENTS FOR THE 1ST LIEUTENANTS"
* Select data
DO sedata8

*Separate Deck and Engineer Officer
USE assposx INDEX assposx
COPY TO asspos FOR spec = "D"
COPY TO asspost FOR spec = "E"
USE assoffx INDEX assoffx
COPY TO assoff FOR spec = "D"
COPY TO assofft FOR spec = "E"
CLOSE DATABASES

ERASE assposx.dbf
ERASE assposx.ndx
ERASE assoffx.dbf
ERASE assoffx.ndx

* Process deck specialty
@ 3,2 SAY "PROCESS DATA DECK OFFICERS"
USE asspos
INDEX ON rank + clayear + claorder TO asspos
USE assoff
INDEX ON rank + clayear + claorder TO assoff
CLOSE DATABASES

@ 4,2 SAY "a. MEDIUM-TO-HIGHER STAGE"
STORE "        " TO tdate
@ 18,2 SAY "ENTER DUE DATE ==>" GET tdate;
                    PICTURE "99/99/99"
READ
@ 18,1 CLEAR TO 18,78
DO rohmh

@ 5,2 SAY "b. MEDIUM-TO-MEDIUM STAGE"
tdate = "        "
@ 18,2 SAY "ENTER DUE DATE ==>" GET tdate;
                       PICTURE "99/99/99"
READ
@ 18,1 CLEAR TO 18,78
DO roelmm
DO romm
DO rohmm
```

```
@ 6,2 SAY "c. LOWER-TO-MEDIUM STAGE"
tdate = "        "
@ 18,1 SAY "ENTER DUE DATE ==>" GET tdate;
                              PICTURE "99/99/99"
READ
@ 18,1 CLEAR TO 18,78
DO rommlm
DO rohlm

* Check data
DO chdata

ERASE asspos.dbf
ERASE asspos.ndx
ERASE assoff.dbf
ERASE assoff.ndx

* process engineer specialty
CLEAR
@ 3,2 SAY "PROCESS DATA ENGINEER OFFICERS"
RENAME asspost.dbf TO asspos.dbf
RENAME assofft.dbf TO assoff.dbf
USE asspos
INDEX ON rank + clayear + claorder TO asspos
USE assoff
INDEX ON rank + clayear + claorder TO assoff
CLOSE DATABASES

@ 4,2 SAY "a. MEDIUM-TO-HIGHER STAGE"
STORE "        " TO tdate
@ 18,2 SAY "ENTER DUE DATE ==>" GET tdate;
                           PICTURE "99/99/99"
READ
@ 18,1 CLEAR TO 18,78
DO rohmh

@ 5,2 SAY "b. MEDIUM-TO-MEDIUM STAGE"
tdate = "        "
@ 18,2 SAY "ENTER DUE DATE ==>" GET tdate;
                              PICTURE "99/99/99"
READ
@ 18,1 CLEAR TO 18,78
DO roelmm
DO romm
DO rohmm

@ 6,2 SAY "c. LOWER-TO-MEDIUM STAGE"
tdate = "        "
@ 18,2 SAY "ENTER DUE DATE ==>" GET tdate;
                              PICTURE "99/99/99"
READ
```

```
@ 18,1 CLEAR TO 18,78
DO rommlm
DO rohlm

* Check data
DO chdata

@ 2,2 SAY "ASSIGNMENT PROCESS TERMINATED"
@ 3,2 SAY "RESULTS ARE PROVIDED THROUGH"
@ 4,2 SAY "LIST AND REPORT MENU"
CLOSE DATABASES

* Update USERLOG data
STORE "ASSIGNMENT" TO dbfunc
STORE "ASSIGN8" TO dbprog
DO userspy

CLOSE DATABASES
ERASE asspos.dbf
ERASE asspos.ndx
ERASE assoff.dbf
ERASE assoff.ndx
RETURN

EOF: ASSIGN8.PRG
```

```
***  PROGRAM SEDATA8
* This program creates the data sources ASSPOS, ASSOFF
* for the assignment process of 1st Lieutenants

SELECT 1
USE officer
SELECT 2
USE power
SELECT 3
USE organic
SELECT 4
USE fleunit

@ 3,2 SAY "SELECT DATA"
@ 4,2 SAY "a. ASSIGNED POSITIONS"
SELECT 3
COPY TO selorg1 FOR orgrank = "8"
SELECT 5
USE selorg1
JOIN WITH D TO orfl1 FOR unit = D->unit
SELECT 6
USE orfl1
JOIN WITH B TO orflpo1 FOR unit = B->unit .AND.;
                           duty = B->duty

SELECT 7
USE orflpo1
JOIN WITH A TO assposx FOR serno = A->serno;
  FIELDS unit, duty, orgrank, orgspec,;
         enrdate, futype, comname, serno,;
         A->rank, A->spec,;
         A->clayear, A->claorder, A->lpdate

SELECT 8
USE assposx
GO TOP
DO WHILE .NOT. EOF()
   IF enrdate > CTOD("07/31/88") - 330;
           .AND. YEAR(lpdate) # YEAR(DATE())
      DELETE
   ENDIF
   SKIP
ENDDO
PACK

SELECT 6
GO TOP
DO WHILE .NOT. EOF()
   SELECT 7
   GO TOP
```

```
   LOCATE FOR .NOT. DELETED() .AND.;
          unit = F->unit .AND. duty = F->duty
   IF EOF()
      SELECT 8
      APPEND BLANK
      REPLACE unit WITH F->unit
      REPLACE duty WITH F->duty
      REPLACE orgrank WITH F->orgrank
      REPLACE orgspec WITH F->orgspec
      REPLACE futype WITH F->futype
      REPLACE comname WITH F->comname
      REPLACE rank WITH F->orgrank
      REPLACE spec WITH F->orgspec
   ENDIF
   SELECT 6
   SKIP
ENDDO

SELECT 8
INDEX ON spec + rank + clayear + claorder;
         TO assposx
CLOSE DATABASES
ERASE selorg1.dbf
ERASE orfl1.dbf
ERASE orflpo1.dbf

SELECT 1
USE officer
SELECT 2
USE power
SELECT 3
USE organic
SELECT 4
USE fleunit

@ 5,2 SAY "b. ASSIGNING OFFICERS"
SELECT 1
COPY TO seloff1 FOR rank = "8"
SELECT 5
USE seloff1
JOIN WITH B TO ofpo1 FOR serno = B->serno
SELECT 6
USE ofpo1
JOIN WITH D TO assoffx FOR unit = D->unit;
     FIELDS serno, rank, spec, clayear, claorder,;
            lpdate, unit, duty, enrdate,;
            D->futype, D->comname

SELECT 7
USE assoffx
GO TOP
```

```
DO WHILE .NOT. EOF()
   IF enrdate > CTOD("07/31/88") - 330;
           .AND. YEAR(lpdate) # YEAR(DATE())
      DELETE
   ENDIF
   SKIP
ENDDO
PACK
CLOSE DATABASES

SELECT 1
USE study
SELECT 2
USE oof
SELECT 3
USE seloffl
SELECT 4
USE assoffx

SELECT 3
JOIN WITH A TO come1 FOR serno = A->serno;
     FIELDS serno, rank, spec, clayear, claorder,;
            lpdate, A->unit, A->duty, A->enrdate
SELECT 3
JOIN WITH B TO come2 FOR serno = B->serno;
     FIELDS serno, rank, spec, clayear, claorder,;
            lpdate, B->unit, B->duty, B->enrdate

SELECT 4
APPEND FROM come1
REPLACE ALL futype WITH "3" FOR duty = "STU"
REPLACE ALL comname WITH "EDU" FOR duty = "STU"
APPEND FROM come2
REPLACE ALL futype WITH "4" FOR duty = "NFD"
REPLACE ALL comname WITH "OUT" FOR duty = "NFD"
SELECT 4
INDEX ON spec + rank + clayear + claorder;
         TO assoffx

CLOSE DATABASES
ERASE seloffl.dbf
ERASE ofpol.dbf
ERASE come1.dbf
ERASE come2.dbf
RETURN

EOF: SELDATA8.PRG
```

```
* PROGRAM CHDATA
* This program checks the final data

CLEAR
SELECT 1
USE assment
SELECT 2
USE assoff INDEX assoff
SELECT 3
USE asspos INDEX asspos

@ 3,2 SAY "CHECK DATA"
STORE .F. TO posdone
STORE .F. TO offdone
SELECT 2
GO TOP
LOCATE FOR .NOT. DELETED()
IF EOF()
   STORE .T. TO offdone
ENDIF
SELECT 3
GO TOP
LOCATE FOR .NOT. DELETED()
IF EOF()
   STORE .T. TO posdone
ENDIF

IF offdone .AND. posdone
   @ 2,2 SAY "ALL GOOD, FINISH GOOD"
   @ 3,2 SAY "NO AVAILABLE POSITIONS"
   @ 4,2 SAY "NO AVAILABLE OFFICERS"
   DO DELAY
   @ 18,1 CLEAR TO 20,78
ENDIF

IF .NOT. offdone
   STORE '0' TO answer
   SELECT 2
   GO TOP
   LOCATE FOR .NOT. DELETED()
   DO WHILE .NOT. EOF()
      CLEAR
      @ 2,2 SAY "THERE ARE STILL AVAILABLE OFFICERS"
      @ 3,2 SAY "1. ASSIGN IN FC"
      @ 4,2 SAY "2. ASSIGN IN HGS"
      @ 6,2 SAY "ENTER YOUR SELECTION ==>";
            GET answer
      READ
      IF answer = '1'
         DELETE
         SELECT 1
```

```
            APPEND BLANK
            REPLACE serno WITH B->serno
            REPLACE assunit WITH "FCS"
            REPLACE assduty WITH "SPARE"
            REPLACE assdate with CTOD(tdate)
         ELSE
            DELETE
            SELECT 1
            APPEND BLANK
            REPLACE serno WITH B->serno
            REPLACE assunit WITH "HGS"
            REPLACE assduty WITH "NFD"
            REPLACE assdate with CTOD(tdate)
         ENDIF
         SELECT 2
         CONTINUE
      ENDDO
   ENDIF

   IF .NOT. posdone
      CLEAR
      @ 2,2 SAY [THERE ARE STILL AVAILABLE POSITIONS]
      CLEAR
      SELECT 3
      GO TOP
      LOCATE FOR .NOT. DELETED()
      DO WHILE .NOT. EOF()
         DISPLAY unit, duty, orgrank, orgspec
         DO delay
         DELETE
         CONTINUE
      ENDDO
   ENDIF
   CLOSE DATABASES
   RETURN

   EOF: CHEDATA.PRG
```

```
***  PROGRAM ROHMH
* This program performs the assignments according to
* the rule of hierarchy in medium-to-higher stage

SELECT  1
USE assment
SELECT 2
USE assoff INDEX assoff
SELECT 3
USE asspos INDEX asspos

SELECT 3
GO TOP
DO WHILE .NOT. EOF()
   STORE .F. TO ok1
   IF .NOT. DELETED() .AND. rank # orgrank
      SELECT 2
      GO TOP
      LOCATE FOR .NOT. DELETED()
      DO WHILE .NOT. EOF()
         IF comname = C->comname .AND. duty = C->duty
            CONTINUE
         ELSE
           DELETE
           SELECT 1
           APPEND BLANK
           REPLACE serno WITH B->serno
           REPLACE assunit WITH C->unit
           REPLACE assduty WITH C->duty
           REPLACE assdate WITH CTOD(tdate)
           STORE .T. TO ok1
           EXIT
        ENDIF
     ENDDO
   ENDIF
   SELECT 3
   IF ok1
      DELETE
   ENDIF
   SKIP
ENDDO
CLOSE DATABASES
RETURN

* EOF: ROHMH.PRG
```

```
*** PROGRAM ROELMM
* This program performs the assignments according to
* the rule of equivalent levels in medium-to-medium stage

SELECT  1
USE assment
SELECT 2
USE assoff INDEX assoff
SELECT 3
USE asspos INDEX asspos

SELECT 3
GO TOP
DO WHILE .NOT. EOF()
   STORE .F. TO ok2
   IF .NOT. DELETED()
      SELECT 2
      GO TOP
      LOCATE FOR .NOT. DELETED();
             .AND. serno = C->serno;
             .AND. YEAR(lpdate) = YEAR(DATE()) - 1
      IF .NOT. EOF()
         DELETE
         STORE .T. TO ok2
      ENDIF
   ENDIF
   SELECT 3
   IF ok2
      DELETE
   ENDIF
   SKIP
ENDDO
CLOSE DATABASES
RETURN

* EOF: ROELMM.PRG
```

```
*** PROGRAM ROMM
* This program performs the assignments according to
* the rule of minimum movements medium-to-medium stage

SELECT  1
USE assment
SELECT 2
USE assoff INDEX assoff
SELECT 3
USE asspos INDEX asspos

SELECT 3
GO TOP
DO WHILE .NOT. EOF()
   STORE .F. TO ok3
   IF .NOT. DELETED()
      SELECT 2
      GO TOP
      LOCATE FOR .NOT. DELETED();
             .AND. unit = C->unit;
             .AND. YEAR(lpdate) # YEAR(DATE())
      IF .NOT. EOF()
         IF serno = C->serno
            DELETE
         ELSE
            DELETE
            SELECT 1
            APPEND BLANK
            REPLACE serno WITH B->serno
            REPLACE assunit WITH C->unit
            REPLACE assduty WITH C->duty
            REPLACE assdate WITH CTOD(tdate)
         ENDIF
         STORE .T. TO ok3
      ENDIF
   ENDIF
   SELECT 3
   IF ok3
      DELETE
   ENDIF
   SKIP
ENDDO
CLOSE DATABASES
RETURN

EOF: ROMM.PRG
```

```
*** PROGRAM ROHMM
* This program performs the assignments according to
* the rule of normal hierarchy in medium-to-medium stage

SELECT  1
USE assment
SELECT 2
USE assoff INDEX assoff
SELECT 3
USE asspos INDEX asspos

SELECT 3
GO TOP
DO WHILE .NOT. EOF()
   STORE .F. TO ok1
   IF .NOT. DELETED()
      SELECT 2
      GO TOP
      LOCATE FOR .NOT. DELETED();
             .AND. YEAR(lpdate) # YEAR(DATE())
      IF .NOT. EOF()
         DELETE
         SELECT 1
         APPEND BLANK
         REPLACE serno WITH B->serno
         REPLACE assunit WITH C->unit
         REPLACE assduty WITH C->duty
         REPLACE assdate WITH CTOD(tdate)
         STORE .T. TO ok1
      ENDIF
   ENDIF
   SELECT 3
   IF ok1
      DELETE
   ENDIF
   SKIP
ENDDO
CLOSE DATABASES
RETURN

EOF: ROHMM.PRG
```

```
*** PROGRAM ROMMLM
* This program performs the assignments according to
* the rule of minimum movements in lower-to-medium stage

SELECT 1
USE assment
SELECT 2
USE assoff INDEX assoff
SELECT 3
USE asspos INDEX asspos

SELECT 3
GO TOP
DO WHILE .NOT. EOF()
   STORE .F. TO ok3
   IF .NOT. DELETED()
      SELECT 2
      GO TOP
      LOCATE FOR .NOT. DELETED();
             .AND. unit = C->unit;
             .AND. YEAR(lpdate) = YEAR(DATE())
      IF .NOT. EOF()
         IF serno = C->serno
            DELETE
         ELSE
            DELETE
            SELECT 1
            APPEND BLANK
            REPLACE serno WITH B->serno
            REPLACE assunit WITH C->unit
            REPLACE assduty WITH C->duty
            REPLACE assdate WITH CTOD(tdate)
         ENDIF
         STORE .T. TO ok3
      ENDIF
   ENDIF
   SELECT 3
   IF ok3
      DELETE
   ENDIF
   SKIP
ENDDO
CLOSE DATABASES
RETURN

EOF: ROMMLM.PRG
```

```
*** PROGRAM ROHLM
* This program performs the assignments according to
* the rule of hierarchy in lower-to-medium stage

SELECT 1
USE assment
SELECT 2
USE assoff INDEX assoff
SELECT 3
USE asspos INDEX asspos

SELECT 3
GO TOP
DO WHILE .NOT. EOF()
   STORE .F. TO ok1
   IF .NOT. DELETED()
      SELECT 2
      GO TOP
      LOCATE FOR .NOT. DELETED();
             .AND. YEAR(lpdate) = YEAR(DATE())
      IF .NOT. EOF()
         DELETE
         SELECT 1
         APPEND BLANK
         REPLACE serno WITH B->serno
         REPLACE assunit WITH C->unit
         REPLACE assduty WITH C->duty
         REPLACE assdate WITH CTOD(tdate)
         STORE .T. TO ok1
      ENDIF
   ENDIF
   SELECT 3
   IF ok1
      DELETE
   ENDIF
   SKIP
ENDDO
CLOSE DATABASES
RETURN

EOF: ROHLM.PRG
```

D. LISTS AND REPORTS PROGRAMS

```
*** PROGRAM LISREP
* This program controls the lists and reports

SET COLOR TO W/B, G/R, BG
CLEAR
STORE .F. TO stop
PUBLIC lrcode
DO WHILE .NOT. STOP
  DO lrmenu
  DO CASE
    CASE lrcode = 0
      RETURN
    CASE lrcode = 1
      DO list1
      STORE .F. TO stop
    CASE lrcode = 2
      DO list2
      STORE .F. TO stop
    CASE lrcode = 3
      DO list3
      STORE .F. TO stop
    CASE lrcode = 4
      DO list4
      STORE .F. TO stop
    CASE lrcode = 5
      DO list5
      STORE .F. TO stop
    CASE lrcode = 6
      DO report1
      STORE .F. TO stop
    CASE lrcode = 7
      DO report2
      STORE .F. TO stop
  ENDCASE
ENDDO
RETURN

* EOF: LISREP.PRG
```

```
*** PROGRAM LRMENU
* This program desplays the lists and reports menu

CLEAR
PUBLIC lrcode
STORE 0 TO lrcode
@  2,0 TO 21,79 DOUBLE
@  4,1 TO  4,78 DOUBLE
@ 19,1 TO 19,78 DOUBLE
SET COLOR TO N*/BR
@  3,24 SAY [L I S T S   A N D   R E P O R T S]
SET COLOR TO N/BR
@  6,20 SAY [1. LIST OF ASSIGNMENTS OF A REQUESTED RANK]
@  7,20 SAY [2. LIST OF OFFICERS OF A REQUESTED UNIT   ]
@  8,20 SAY [3. LIST OF OFFICERS OF A REQUESTED DUTY   ]
@  9,20 SAY [4. LIST OF OFFICERS OF A REQUESTED RANK   ]
@ 10,20 SAY [5. LIST OF OFFICERS IN A REQUESTED ORDER  ]
@ 11,20 SAY [6. REPORT OF OFFICER'S CAREER HISTORY     ]
@ 12,20 SAY [7. REPORT OF OFFICER'S CURRENT STATUS     ]
@ 13,20 SAY [..........................................]
@ 15,20 SAY [0. EXIT AND RETURN TO MAIN MENU           ]
SET COLOR TO W+/BR
@ 20,25 SAY 'ENTER YOUR SELECTION ==>' ;
        GET lrcode PICTURE "9" RANGE 0,7
READ
SET COLOR TO W/B, G/R, BG
RETURN

* EOF: LRMENU.PRG
```

```
*** PROGRAM LIST1
* This program creates output list of assignments of a
* requested rank

CLEAR
@ 1,0 TO 11,40 DOUBLE
@ 3,1 TO 3,39 DOUBLE
@ 2,4 SAY "LISTS OF SCHEDULING ASSIGNMENTS"
@ 9,1 TO 9,39 DOUBLE
STORE " " TO ans
@ 10,2 SAY "DO YOU NEED CODES (Y/N) ==>" GET ans;
       PICTURE "A"
READ
IF UPPER(ans) = "Y"
   DO rscodes
ENDIF
@ 10,1 CLEAR TO 10,39

STORE "0" TO lrrank
@ 10,2 SAY "ENTER REQUESTED RANK CODE ==>" GET lrrank;
            PICTURE "9"
READ
@ 0,42 CLEAR TO 21,79
@ 10,1 CLEAR TO 10,39
@ 10,2 SAY "PROCESSING IN PROGRESS"
USE officer
INDEX ON rank TO officer
USE assment
INDEX ON serno TO assment

SELECT 1
USE officer INDEX officer
SELECT 2
USE power
SELECT 3
USE study
SELECT 4
USE oof
SELECT 5
USE assment INDEX assment

SELECT 1
COPY TO lst1 FOR rank = lrrank

SELECT 6
USE lst1
JOIN WITH B TO lst21 FOR serno = B->serno
SELECT 6
JOIN WITH C TO lst22 FOR serno = C->serno
SELECT 6
JOIN WITH D TO lst23 FOR serno = D->serno
```

```
SELECT 7
USE lst21
APPEND FROM lst22
APPEND FROM lst23

IF lrrank = "9"
  APPEND FROM officer FOR VAL(clayear) = YEAR(DATE())
  REPLACE ALL unit WITH "NACADEMY";
          FOR VAL(clayear) = YEAR(DATE())
ENDIF

SELECT 7
JOIN WITH E TO lart FOR serno = E->serno
CLOSE DATABASES

SELECT 1
USE dvrank
SELECT 2
USE lart
JOIN WITH A TO lar FOR rank = A->rank;
     FIELDS serno, name, rank, spec, clayear,;
            claorder, unit, duty, enrdate,;
            assunit, assduty, assdate, A->rankname
SELECT 4
USE lar
INDEX ON spec + rank + clayear + claorder TO lar

STORE .T. TO lamenu
DO WHILE lamenu
   STORE 0 TO lamn
   @ 4, 2 SAY [1. LIST ASSIGNMENTS DECK SPECIALTY]
   @ 5, 2 SAY [2. LIST ASSIGNMENTS ENGINEER SPECIALTY]
   @ 7, 2 SAY [0. NO LISTS - EXIT]
   @ 10,1 CLEAR TO 10,39
   @ 10,2 SAY "SWITCH ON YOUR PRINTER"
   DO delay
   @ 10,1 CLEAR TO 10,39
   @ 10,2 SAY "ENTER YOUR SELECTION ==>" GET lamn;
       PICTURE "9" RANGE 0,2
   READ
   DO CASE
      CASE lamn = 0
         lamenu = .F.
      CASE lamn = 1
         SET CONSOLE OFF
         SET PRINT ON
         REPORT FORM lar1 FOR spec = "D"
         REPORT FORM lar2 FOR spec = "D"
         SET PRINT OFF
         SET CONSOLE ON
```

```
        CASE lamn = 2
           SET CONSOLE OFF
           SET PRINT ON
           REPORT FORM lar1 FOR spec = "E"
           REPORT FORM lar2 FOR spec = "E"
           SET PRINT OFF
           SET CONSOLE ON
     ENDCASE
  ENDDO
  CLOSE DATABASES
  @ 10,1 CLEAR TO 10,39
  @ 10,2 SAY "PROCESS FINISH"
  STORE "LISREP" TO dbfunc
  STORE "LIST1" TO dbprog
  DO userspy

  CLOSE DATABASES
  ERASE lst1.dbf
  ERASE lst21.dbf
  ERASE lst22.dbf
  ERASE lst23.dbf
  ERASE lart.dbf
  ERASE lar.dbf
  ERASE lar.ndx
  ERASE officer.ndx
  ERASE assment.ndx
  RETURN

  EOF: LIST1.PRG
```

```
*** PROGRAM REPORT2
* This program provides output report of an officer's
* current status

CLEAR
@ 0,0 TO 21,79 DOUBLE
@ 2,1 TO 2,78 DOUBLE
@ 1,5 SAY "OFFICER'S CURRENT STATUS REPORT"
@ 19,1 TO 19,78 DOUBLE

USE power
INDEX ON serno TO power
USE study
INDEX ON serno TO study
USE oof
INDEX ON serno TO oof
USE officer
INDEX ON serno TO officer
CLOSE DATABASES

SELECT 2
USE officer INDEX officer
SELECT 3
USE power INDEX power
SELECT 4
USE study INDEX study
SELECT 5
USE oof INDEX oof
SELECT 6
USE dvrank
SELECT 7
USE nomhist

STORE DATE()                    TO tdate
STORE "     "                   TO tserno
STORE "          "              TO tunit
STORE "    "                    TO tduty
STORE "        "                TO tenrdate

@ 20,5 SAY "ENTER SERIAL NUMBER ==>" GET tserno;
          PICTURE "99999"
READ
@ 20,1 CLEAR TO 20,78

STORE "0" TO answer
@ 3,2 SAY "1. PRINTER OUTPUT"
@ 4,2 SAY "2. DESPLAY SCREEN"
@ 20,1 SAY "ENTER YOUR SELECTION" GET answer;
              PICTURE "9"
READ
```

```
SELECT 2
FIND &tserno
IF .NOT. EOF()
   SELECT 3
   FIND &tserno
     IF EOF()
        SELECT 4
        FIND &tserno
        IF EOF()
           SELECT 5
           FIND &tserno
        ENDIF
     ENDIF
   STORE unit TO tunit
   STORE duty TO tduty
   STORE enrdate TO tenrdate
   SELECT 6
   LOCATE FOR rank = B->rank
   SELECT 7
   LOCATE FOR serno = B->serno

   IF answer = "1"
      @ 20,1 SAY "SWITCH ON YOUR PRINTER"
      DO delay
      SET PRINT ON
   ENDIF
CLEAR

?
?
?
? "                                              HNGS/FC"
? "                                              -------"
? "                                              DATE:",tdate
?
?
? "                 CURRENT OFFICER'S STATUS REPORT"
? "                 ==============================="
?
? "               SERIAL NUMBER        : ", B->serno
? "               NAME                 : ", B->name
? "               RANK                 : ", F->rankname
? "               SPECIALTY            : ", B->spec
? "               NOMINATION DATE      : ", G->hdate
? "               LAST PROMOTION DATE  : ", B->lpdate
? "               CLASS ORDER          : ", B->claorder
? "               MARITAL STATUS       : ", B->mstatus
? "               UNIT                 : ", tunit
? "               DUTY                 : ", tduty
? "               ENROLMENT DATE       : ", tenrdate
?
```

```
   IF answer = "1"
      SET PRINT OFF
      SET CONSOLE ON
   ENDIF
ELSE
   @ 21,1 SAY " OFFICER DOES NOT EXIST"
   DO delay
ENDIF
CLOSE DATABASES

STORE "LISREP" TO dbfunc
STORE "REPORT2" TO dbprog
do userspy
CLOSE DATABASES

ERASE power.ndx
ERASE study.ndx
ERASE oof.ndx
ERASE officer.ndx
RETURN

* EOF: REPORT2.PRG
```

E. MISCELLANEUS PROGRAMS

```
*** PROGRAM USERSPY
* This program records log-data into USERLOG file

PUBLIC psw
SELECT 4
USE userpass
SELECT 5
use userlog

SELECT 4
GO TOP
LOCATE FOR password = UPPER(psw)
SELECT 5
APPEND BLANK
REPLACE username WITH D-> username
REPLACE function WITH dbfunc
REPLACE progname WITH dbprog
REPLACE logdate WITH DATE()
REPLACE logtime WITH TIME()
CLOSE DATABASES
RETURN

EOF: USERSPY.PRG
```

```
*** PROGRAM DELAY
* This program provides a small delay necessary for
* displaying various program messages on the screen

STORE 0 TO k
DO WHILE k < 100
STORE k + 1 TO k
ENDDO
RETURN

EOF: DELAY.PRG
```

```
***  PROGRAM RSCODES
* This program displays on the screen the rank
* and specialty codes

@  0,45 TO 13,74 DOUBLE
@  1,50 SAY [RANK CODES]
@  2,46 TO 2,73 DOUBLE
@  3,47 SAY [9 = ENSIGN          (ENS)  ]
@  4,47 SAY [8 = 1ST LIEUTENANT  (1LT)  ]
@  5,47 SAY [7 = LIEUTENANT      (LT)   ]
@  6,47 SAY [6 = LT COMMANDER    (LCDR) ]
@  7,47 SAY [5 = COMMANDER       (CDR)  ]
@  8,47 SAY [4 = CAPTAIN         (CAPT) ]
@  9,47 SAY [3 = COMMODORE       (COMD) ]
@ 10,47 SAY [2 = REAR ADMIRAL    (RADM) ]
@ 11,47 SAY [1 = VICE ADMIRAL    (VADM) ]
@ 12,47 SAY [0 = ADMIRAL         (ADM)  ]

@ 14,45 TO 21,74 DOUBLE
@ 15,50 SAY [SPECIALTY CODES]
@ 16,46 TO 16,73 DOUBLE
@ 17,47 SAY [D = DECK     ]
@ 18,47 SAY [E = ENGINEER]
@ 19,47 SAY [M = MEDICAL ]
@ 20,47 SAY [S = SUPPLY  ]
RETURN

EOF: RSCODES.PRG
```

F. SAMPLE LISTS AND REPORTS

Page No. 1
09/12/88

LIST OF SCHEDULED ASSIGNMENTS

FOR THE REQUESTED RANK - "FORM 1"

RANK	SPEC	NAME	FROM UNIT	TO UNIT	DUE DATE
1LT	D	ALLEN GEORGE A	KRIEZIS	MIKONIOS	07/11/88
1LT	D	BARBER JEFF G	MIAOULIS	PONTOS	07/11/88
1LT	D	QUICK JIM B	SACHTOURIS	SACHTOURIS	07/11/88
1LT	D	RAMEN HAROL F	APOSTOLIS	APOSTOLIS	07/11/88
1LT	D	VIERA FRANK K	XENOS	NIOVI	07/11/88
1LT	D	RIHOS ARIS H	OKEANOS	KLIO	07/11/88
1LT	D	BARLOT PAUL D	KRIEZIS	KRIEZIS	07/21/88
1LT	D	CRISLER THOMAS B	HGS	VELOS	07/21/88
1LT	D	ITALIS GIORGOS Q	SESCHOOL	MIAOULIS	07/31/88
1LT	D	GALIS PARIS R	SESCHOOL	SACHTOURIS	07/31/88
1LT	D	SPANOS MIMIS S	SESCHOOL	APOSTOLIS	07/31/88
1LT	D	NORVIGAS THOMAS Y	SESCHOOL	XENOS	07/31/88
1LT	D	SUIDAS BEN D	SESCHOOL	OKEANOS	07/31/88
1LT	D	FILANDIS PETER G	SESCHOOL	KRIEZIS	07/31/88
1LT	D	PORTOGAS SPIROS R	SESCHOOL	FCS	07/31/88

Page No. 1
09/12/88

LIST OF SCHEDULED ASSIGNMENTS

FOR THE REQUESTED RANK - "FORM 2"

SERNO	RANK	SPEC	NAME	OLD DUTY	NEW DUTY
82811	1LT	D	ALLEN GEORGE A	CON	OPE
82815	1LT	D	BARBER JEFF G	CON	OPE
82819	1LT	D	QUICK JIM B	ASW	CON
82823	1LT	D	RAMEN HAROL F	ASW	CON
82825	1LT	D	VIERA FRANK K	OPE	EXE
82845	1LT	D	RIHOS ARIS H	OPE	EXE
83805	1LT	D	BARLOT PAUL D	ASW	CON
84860	1LT	D	CRISLER THOMAS B	NFD	CON
85801	1LT	D	ITALIS GIORGOS Q	STU	CON
85802	1LT	D	GALIS PARIS R	STU	ASW
85808	1LT	D	SPANOS MIMIS S	STU	ASW
85812	1LT	D	NORVIGAS THOMAS Y	STU	OPE
85814	1LT	D	SUIDAS BEN D	STU	OPE
85818	1LT	D	FILANDIS PETER G	STU	ASW
85820	1LT	D	PORTOGAS SPIROS R	STU	SPAR

LIST OF SCHEDULED ASSIGNMENTS

FOR THE REQUESTED RANK - "FORM 1"

RANK	SPEC	NAME	FROM UNIT	TO UNIT	DUE DATE
1LT	E	ABLAN BRUNO W	APOSTOLIS	MIKONIOS	08/11/88
1LT	E	FURMAN JOSON U	XENOS	SACHTOURIS	08/11/88
1LT	E	ABAL JOHN F	MIAOULIS	OKEANOS	08/11/88
1LT	E	GARMONY THEAD G	STARAKIS	KRIEZIS	08/11/88
1LT	E	PAKISTOS TED F	SESCHOOL	APOSTOLIS	07/31/88
1LT	E	IRANIS GEORGE H	SESCHOOL	XENOS	07/31/88
1LT	E	KINEZIS PETER L	SESCHOOL	MIAOULIS	07/31/88
1LT	E	KORIOS DIM R	SESCHOOL	STARAKIS	07/31/88
1LT	E	IAPONES THOMAS I	SESCHOOL	HGS	07/31/88

Page No. 1
09/12/88

LIST OF SCHEDULED ASSIGNMENTS

FOR THE REQUESTED RANK - "FORM 2"

SERNO	RANK	SPEC	NAME	OLD DUTY	NEW DUTY
08202	1LT	E	ABLAN BRUNO W	DAM	ENG
08206	1LT	E	FURMAN JOSON U	ENG	DAM
08218	1LT	E	ABAL JOHN F	DAM	ENG
08305	1LT	E	GARMONY THEAD G	ENG	DAM
08501	1LT	E	PAKISTOS TED F	STU	DAM
08503	1LT	E	IRANIS GEORGE H	STU	ENG
08506	1LT	E	KINEZIS PETER L	STU	DAM
08508	1LT	E	KORIOS DIM R	STU	ENG
08510	1LT	E	IAPONES THOMAS I	STU	NFD

HNGS/FC

DATE: 09/10/88

CURRENT OFFICER'S STATUS REPORT

SERNO NUMBER	:	79718
NAME	:	RADASON PETER C
RANK	:	LT
SPECIALTY	:	D
NOMINATION DATE	:	06/17/79
LAST PROMOTION DATE	:	06/11/86
CLASS ORDER	:	18
MARITAL STATUS	:	M
UNIT	:	SACHTOURIS
DUTY	:	OPE
ENROLMENT DATE	:	06/20/86

LIST OF REFERENCES

1. Kroenke, David, Database Processing, Second Edition, Science Research Associates, Inc., 1983.

2. Ullman, Jeffrey D., Principles of Database Systems, Second Edition, Computer Science Press, Inc., 1982.

3. Korth, H. F., Silberschatz, A., Database System Concepts, McGraw-Hill, Inc., 1986.

4. Senn, Janes, Analysis and Design of Information Systems, McGraw-Hill, Inc., 1984.

5. Fairley, Richard, Software Engineering Concepts, McGraw-Hill, Inc., 1985.

6. Hawryszkiewycz, T. I., Database Analysis and Design, SRA, Inc., 1984.

7. Kroenke, David, and Dolan, Kathleen, Database Processing, Third Edition, SRA, Inc., 1988.

8. Howe, D. R., Data Analysis for Data Base Design, Edward Arnold Publishers, Ltd., 1983.

9. Jane's Fighting Ships 1987-1988, Jane's Publishing, Inc., 1987.

10. Simson, Alan, Understanding dBASE III PLUS, SYBEX, Inc., 1986.

11. Jones, Edward, Using dBASE III PLUS, McGraw-Hill, Inc., 1987.

12. Yao, S. B., Principles of Database Design, Prentice-Hall, Inc., 1985.

13. Tanimoto, Steven, L., The Elements of Artificial Intelligence, Computer Science Press, Inc., 1987.

14. Turban, Efraim, Decision Support and Expert Systems, Macmillan Publishing Company, 1988.

15. Bertsekas, Dimitri, and Gallager, Robert, Data Networks, Prentice-Hall, Inc., 1987.

16. Modell, M. E., IEEE 1985 <u>International Conference on Entity-Relationship Approach: The Entity-Relationship Approach as a Tool for Application Analysis</u>. IEEE Computer Society Press, 1985.

17. Ling, T. W., IEEE 1985 <u>International Conference on Entity-Relationship Approach: A Normal Form for Entity-Relation Diagram</u>. IEEE Computer Society Press, 1985.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Director, Information Systems (OP-945) Office of the Chief of Naval Operations Navy Department Washington, DC 20350-2000	1
4.	Superintendent, Naval Postgraduate School Computer Technology Programs, Code 37 Monterey, California 93943-5000	1
5.	Major John B. Isset, Code 54IS Department of Administrative Science Naval Postgraduate School Monterey, California 93943-5000	1
6.	Prof. Vincent V. Lum, Code 52LU Department of Computer Science Naval Postgraduate School Monterey, California 93943-5000	1
7.	Hellenic Navy General Staff GEN/B4 Stratopedo Papagou Holargos 15562 Athens, GREECE	5
8.	LT Antonios Makris Hellenic Navy General Staff Stratopedo Papagou Holargos 15562 Athens, GREECE	2